NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

FLOW FIELD MEASUREMENTS USING
HOTWIRE ANEMOMETRY

by

Gregory John Doremus

September 1987

Thesis Advisor: S. Bodapati

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution is Unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (If applicable) 67	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000		7b ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c ADDRESS (City, State, and ZIP Code)

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
FLOW FIELD MEASUREMENTS USING HOTWIRE ANEMOMETRY

12 PERSONAL AUTHOR(S)
Doremus, Gregory J.

13a TYPE OF REPORT Master's Thesis	13b TIME COVERED FROM ______ TO ______	14 DATE OF REPORT (Year Month Day) 1987 September	15 PAGE COUNT 105

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Data Acquisition, Hotwire, Wake Analysis, Wind Tunnel

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

A computer controlled data acquisition system utilizing hotwire anemometry has been designed, built, and installed in the Naval Postgraduate School Low Speed Wind Tunnel Facility. All relevant wind tunnel data is obtained by the use of a computer guided Hewlett Packard data acquisition system and a Para-Scientific Pressure Computer. Two computer programs were written to coordinate hotwire system calibration with hotwire experimentation. An experiment, Wake Velocity Profile Analyisis and Drag Coefficient Measurement of an Airfoil, was used as a vehicle to test the system. The final output of the data acquisition system, including graphical information, compared favorably with previous results from an older data acquisition system already in use. Drag coefficient output compared very favorably to data supplied by the National Advisory Committee on Aeronautics.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL S. Bodapati	22b TELEPHONE (Include Area Code) 408-646-2854	22c OFFICE SYMBOL 67Bu

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Flow Field Measurements Using Hotwire Anemometry

by

Gregory John Doremus
Lieutenant, United States Navy
B.S., SUNY at Stony Brook, 1980

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the
NAVAL POSTGRADUATE SCHOOL
September 1987

ABSTRACT

A computer controlled data acquisition system utilizing hotwire anemometry has been designed, built, and installed in the Naval Postgraduate School Low Speed Wind Tunnel Facility. All relevant wind tunnel data is obtained by the use of a computer guided Hewlett Packard data acquisition system and a Para-Scientific Pressure Computer. Two computer programs were written to coordinate hotwire system calibration with hotwire experimentation. An experiment, Wake Velocity Profile Analyisis and Drag Coefficient Measurement of an Airfoil, was used as a vehicle to test the system. The final output of the data acquisition system, including graphical information, compared favorably with previous results from an older data acquisition system already in use. Drag coefficient output compared very favorably to data supplied by the National Advisory Committee on Aeronautics.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

I would like to thank Jack King and Al McGuire for their invaluable time, insight and assistance. Their friendship and good humor truly made this thesis possible.

I would also like to thank Lt. Jeff Kunkel who always took the time to help iron out innumerable computer related dilemmas.

Special thanks to M. S. Chandrasekhara whose knowledge and expertise in hotwire anemometry were more valuable to me than any reference book.

A very special thank you to Dr. Bodapati who above all, believed in me.

I. INTRODUCTION

It was the goal of this thesis project to design, build and implement a computer operated data acquisition system for the NPS wind tunnel utilizing hotwire anemometry. Preliminary project goals were to interface the new hotwire data acquisition system with the existing pitot-static system for on-line computer analysis and comparison. Additionally, the new computer program must produce hardcopy output of all graphical and tabular data. A final design requirement was storage of all wake velocity profile data for future analysis. Versatility was the foremost design requirement for the data acquisition computer program. The computer program is to have application as a classroom demonstration tool or laboratory introduction to hotwire anemometry. The system can also be used in future research of aerodynamic models or airfoil sections.

To enhance adaptability, the computer program was created in two parts. The first part, program CALIB, performs system set up and calibrates the hotwire. As a test case for the part two application program, an experiment was conducted where the wake velocity profile of a NACA 66(215)-216 airfoil was analyzed.

Both parts of the application program guide the user through each step of the experiment. Raw data, reduced data, and graphical outputs are all tabulated on screen for immediate examination. Hard copy output of all on screen information is also available. The second application program, PART2, provides wake velocity profiles of the airfoil using hotwire anemometry and a pitot-static system for comparison. Drag coefficients are also calculated for the airfoil as a function of air flow velocity and angle of incidence.

II. THEORY AND INSTRUMENTATION

A. HOT WIRE ANEMOMETRY

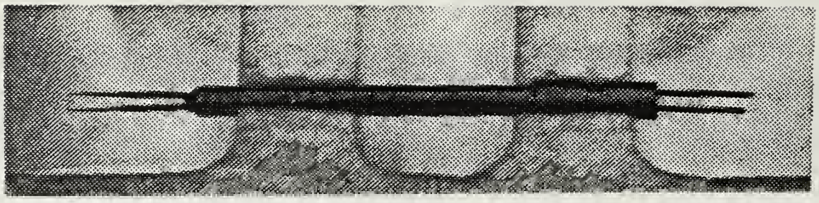
A thermal anemometer is used to measure fluid velocity by responding to changes in heat transfer from an electrically heated probe immersed in a fluid flow. The amount of electrical energy dissipated from the sensor is a measure of the cooling effect of the fluid flowing past the heated sensor. Heat loss from the hotwire probe is dependent on the wire operating temperature, geometry of the wire, thermal properties of the wire, and physical properties of the fluid flow (ie, air temperature, velocity, pressure, and density).

Air flow in the NPS wind tunnel can be considered low speed, steady, incompressible flow. Under these conditions a reasonable assumption can be made that the fluid temperature, density and pressure are non-varying. With this assumption heat loss from the wire sensor can be interpreted as a direct measure of the fluid velocity.

The sensing element, also called the hotwire sensor or probe, is a fine metal wire (4-5 microns in diameter) suspended between two metal supports. A cylindrical hot film sensor consists of a wire shaped electrical insulator called a substrate with a conducting film deposited on the outside surface. Its outward appearance is very similar to that of a hotwire (see Figure 1), the main difference being diameter (about 50 microns for a hot film sensor as compared to 4-5 microns for a hotwire).

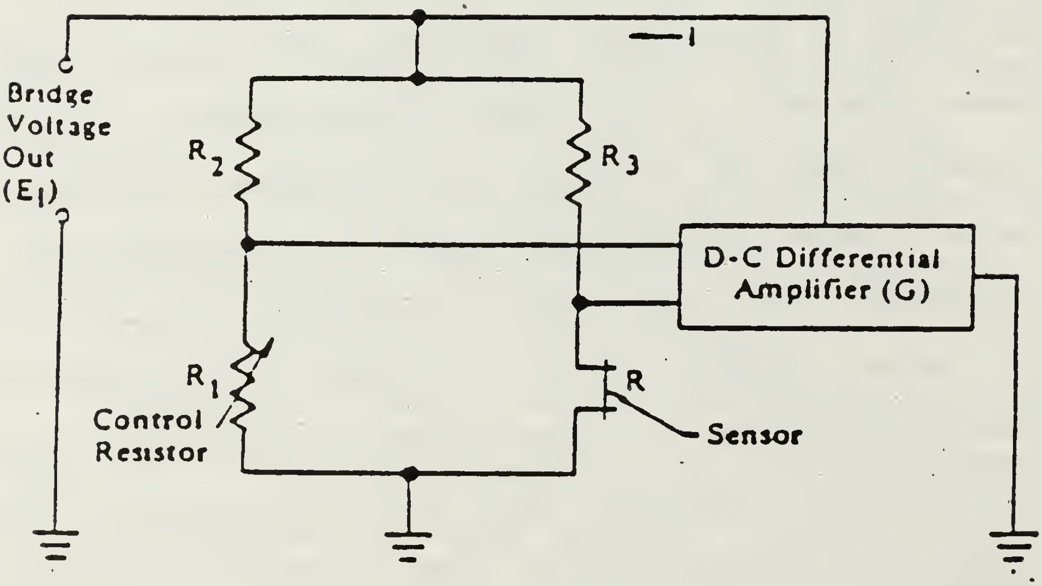
The sensor is mounted on a long thin probe in an effort to be as unobtrusive to the ambient flow as possible. Electrical current to the sensor is supplied by the anemometer, consisting of a Wheatstone Bridge and a control amplifier (see Figure 2). The other resistances in the bridge (R2 and R3) are fixed.

The wheatstone bridge and control amplifier work on a voltage balancing principle. Air flow in the wind tunnel will cool the sensor resulting in a decrease in its resistance. The decreased resistance of the probe lowers the voltage across the wheatstone bridge which in turn lowers the input to the control amplifier. The control amplifier is phase adjusted; a lower input voltage causes a current increase from the amplifier to the bridge. The increased current heats up the sensor which also increases its resistance. When the probe's resistance is back to its pre-set value, voltages across the bridge are balanced and the control amplifier no longer alters current flow through the bridge. If the amplifier gain is set correctly, voltage inputs will not deviate far from the balanced condition. Therefore any change in sensor resistance due to fluid flow fluctuations will be corrected immediately.



TSI Model 1210-20 Hot Film Sensor

Figure 1



Schematic of a Constant Temperature Hotwire System

Figure 2

The control amplifier is able to maintain a voltage balance across the hotwire probe and wheatstone bridge at a rate of almost 200,000 cycles per second. Thus, hotwire anemometry is ideally suited for rapidly changing or turbulent flow regimes. Quick response to velocity fluctuations also provides an accurate, repeatable output when steady flow conditions prevail. The IFA 100 Thermal Anemometer used in this thesis is shown in Figure 3.

B. HEAT TRANSFER IN A HOTWIRE SENSOR

Electrical heat input to a hotwire probe is dissipated in various modes of heat transfer. Radiation, buoyant convection, conduction to the sensor's end supports and forced convection all contribute to sensor cooling in the flow field. Radiative heat loss from a typical hotwire sensor is less than 0.1 percent of the heat input to the probe. In very low density gaseous flows this source of heat loss may be important. Radiative heat losses were considered negligible in this thesis.

Buoyant convection is important in heat loss calculations only at very low wind tunnel speeds. Buoyant convection can be neglected at fluid speeds greater than 5 centimeters/second (about one mile per hour). The primary contributors to heat transfer from a hotwire sensor are conduction to the end supports and forced convection by the fluid flow.

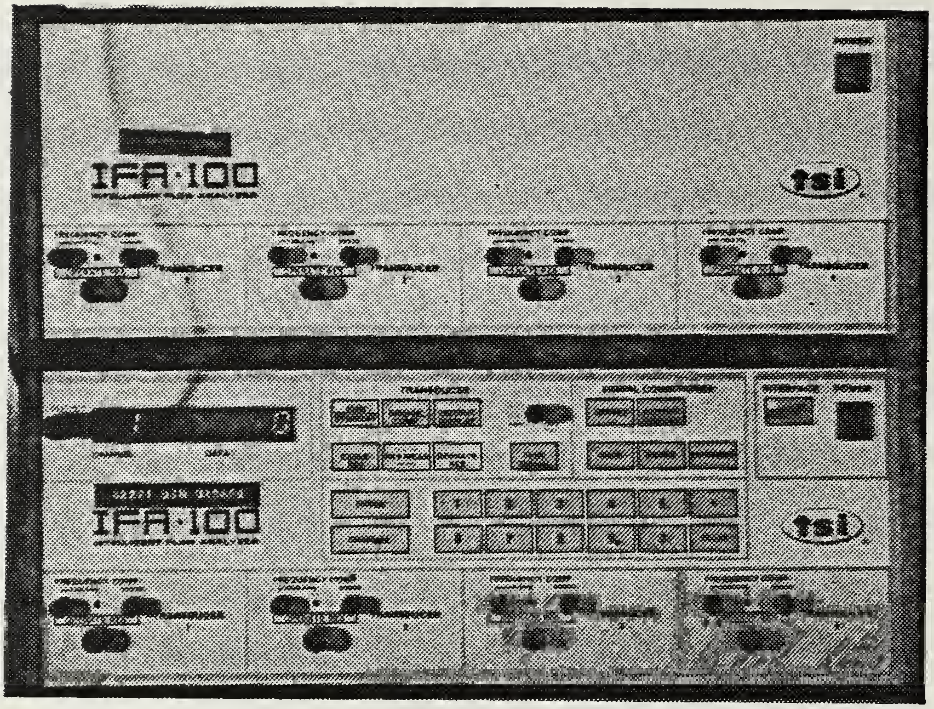
The supports that hold the hotwire sensor to the probe legs are much thicker than the wire sensor. This serves a dual purpose: (1) The supports contribute to the sensor's strength; (2) The supports should not be heated appreciably by an electrical current flowing through the wire sensor. When the sensor is operating in a flow field the temperature along the wire is greatest in the center. The temperature at the center is closest to that of an infinite wire, not affected by the supports. Nearer to the supports the wire temperature decreases until it is very close to the temperature of the ambient flow field. Heat transfer from the sensor to the supports by conduction is a loss and a potential source of error.

It has been experimentally determined that approximately 17 percent of the heat input to a typical hotwire sensor is lost to the supports through conduction. The remaining heat input dissipates through the process of forced convection. These numbers represent large experimental errors if the heat loss modes are not accounted for. Heat conduction to the end supports is the best argument for calibrating the hotwire system with the wind tunnel on (under operating conditions) instead of using universal calibration constants. The calibration curve, if created under operating conditions, will properly account for conduction losses.

C. HP DATA ACQUISITION SYSTEM

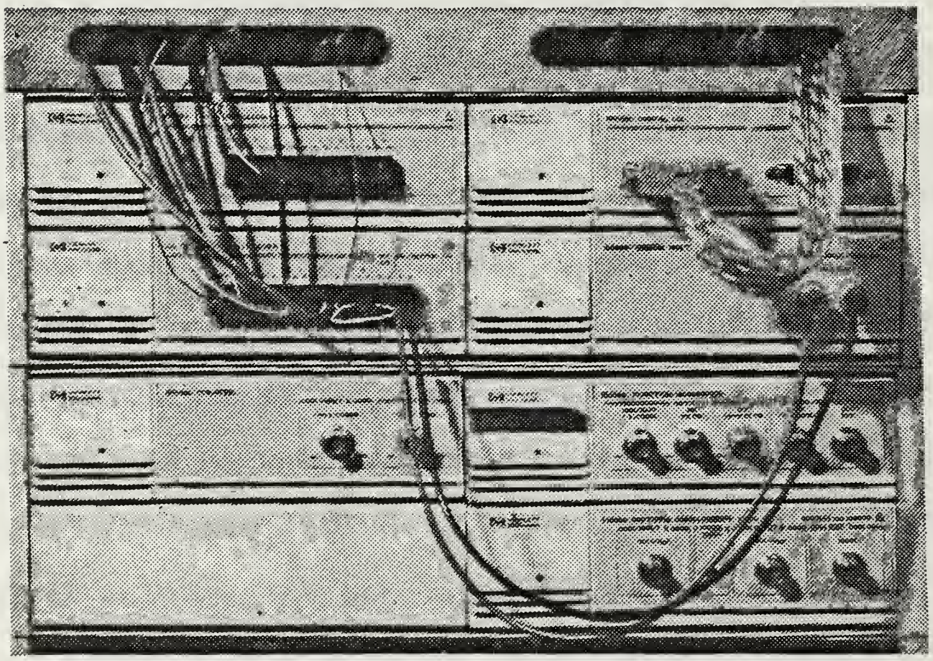
The HP Data Acquisition System consists of a combination of hardware and software that allows the IBM PC-AT computer to act as a fully automated instrumentation system. The individual HP instruments used in this thesis include the Relay Multiplexer, Digital Multimeter, Relay Actuator, and Digital Input Port (see Figure 4). Each of the instruments can be operated manually at the computer screen by means of the mouse control. This is made possible by a software program called the PC Instruments Soft Front Panel. In this mode the instruments' operating controls and functions, as well as digital displays are relayed to the computer screen. Instead of user interaction with the instruments' controls and indicators, each unit is manipulated by the computer mouse controller.

In addition to the manual mode, the HP data acquisition system can operate in the programmed (basic) mode. This method is employed in this thesis. A program is written in the Advanced Basic language (BASICA) that calls up each instrument and its function as necessary. All data acquisition related to the hotwire wind tunnel experiment is accomplished by the use of a BASICA program. User



IFA 100 Thermal Anemometer

Figure 3



HP Data Acquisition System

Figure 4

interaction in the experiment is limited to those tasks not controlled by the HP data acquisition system.

D. INSTRUMENTATION

1. Relay Multiplexer

The Relay Multiplexer is capable of switching between eight input channels of voltage data. Each of the eight inputs consists of two relay contacts (double pole-single throw). The system provides one common output channel which in turn can be read by the Digital Multimeter. Multple relay closures cannot occur; only one input can be switched to the common output at a time. The Relay Multiplexer utilizes break-before-make input switching. The presently selected input channel's signal is opened and interrupted before a new input channel is connected to the common output. Figure 10 in Chapter III shows all data signal paths routed to the Relay Multiplexer.

2. Digital Multimeter

The Digital Multimeter (DMM) can measure + or - DC voltages, AC voltages, and ohms. In this thesis only the DC voltage measurement function is used. Direct current voltages can be read on four ranges varying from 0.2 V to 200 V. An auto range feature also exists where the DMM selects the optimal range for the signal that is being measured.

The DMM automatically converts input analog voltage signals into a digital (or binary) form which can be read by the computer. When the DMM sends its measurement to the computer its value is assigned to a variable, placed in an array, or written to a permanent data file. The DMM has a continuous data sampling rate of 2.5 or 12.5 readings per second. This sampling rate can be altered as desired by a command from the BASICA application program. The slower sampling rate has a higher accuracy (+/- .03% of the input voltage) than the higher sampling rate (+/- .05% of the input voltage). Thus, the slower sampling rate is used throughout the experiments.

When a data reading is called for, the last complete reading obtained by the DMM is output to the computer. Speed of data acquisition is limited to the speed in which the IBM computer carries out each BASICA command. This varies between 15-120 milliseconds per command. When input channels of the Relay Multiplexer are switched a delay of approximately one second is required to allow the circuit to settle before a measurement is taken. The result is a maximum data acquisition rate of approximately ten readings per second when one channel is input and only one reading per second when two or more consecutive channels are input. The HP data acquisition system is essentially a steady state system.

3. Relay Actuator

The Relay Actuator contains eight independently controllable electrical relays. Each of the eight channels consists of one single pole-single throw relay. Unlike the Relay Multiplexer, all eight channels can be set in any configuration (open or closed) simultaneously. The Relay Actuator is used solely in controlling the angle of incidence setting for the airfoil.

4. Digital Input Port

The Digital Input Port is able to read digitized data in four different modes: Binary, Decimal, Hexadecimal, and Octal.Each of these modes can be displayed on the computer screen when the Digital Input Port is interactive through the Soft Front Panels software. When a measurement is called from a BASICA application code however, the output to the computer is always in the Decimal mode with 2's complement and positive logic. In this mode the Digital Input Port assigns a bit weight to each of the input pins that read "high" from the Para-Scientific Pressure Computer. Digital output from the Para-Scientific Pressure Computer is discussed in detail in Chapter II. D. 5.

If the first pin, or most significant bit, reads high the Digital Input Port assigns a negative sign to the output. A typical Decimal value corresponding to a pressure value of 14.73 PSI looks like this: -32766. A subroutine in CALIB and PART2 inputs the Decimal value and scales it to determine which pins read high and which give no input (read low). When scaling is complete the input Decimal value is converted to its numeric equivalent (in this example -32766 becomes 14.73) in pounds per square inch.

The Digital Input Port is set to read input digitized data continuously. The Para-Scientific Pressure Computer updates its pressure reading approximately 3 times per second. Therefore a new pressure reading, in digital form, is sent to the Digital Input Port 3 times per second. When a measurement is called from the BASICA application code of CALIB or PART2, the last complete measurement taken after the measure statement is sent to the computer. This value is an instantaneous measurement from the Para-Scientific Transducer (not RMS). To reduce the chance of spurious inputs, 70 measurements are called for consecutively and averaged in the computer program. Each measurement takes approximately 30 milliseconds so the final pressure reading spans 6 pressure updates from the Para-Scientific Pressure Computer.

5. Para-Scientific Pressure Computer

The Para-Scientific Pressure Computer can accept inputs from up to eight pressure transducers. Two are utilized for data acquisition in this thesis project. The first input (P1) is from a differential pressure transducer. The differential transducer reads pitot-static pressure from the individual total and static probes in the test section.

The static pressure line is split by means of a "T" connector. One line is connected to the differential pressure transducer, the other is connected to an absolute pressure transducer. Static pressure from the absolute transducer makes up the second input (P2) to the Para-Scientific Pressure Computer

The pressure transducers used by the Pressure Computer are factory calibrated. Coefficients are supplied with each transducer that enable the Pressure Computer to convert pressure readings to an output of PSI. The Pressure Computer's output is updated continuously at a rate of approximately 3 readings per second. Output data can be analog or digital (BCD). Digital output is used in this thesis for speed and accuracy.

The Pressure Computer provides a 50 pin digital output to accomodate pressure readings as high as 10,000 PSI with an accuracy to the fifth decimal (0.00001 PSI). The limitation on this system was in the HP Digital Input Port. The Digital Input Port only accepts 16 pins of input data so the 16 most significant data pins for this experiment had to be used. Data taken from the Pressure Computer and input to the Digital Input Port is outlined in Figure 5. The accuracy of this configuration is +/- 0.002 PSI. The Pressure Computer can measure pressures from its two input

Pin #	Value (PSI)	Weight	
1	10.	1	MSB
2	1.	8	
3	1.	4	
4	1.	2	
5	1.	1	
6	.1	8	
7	.1	4	
8	.1	2	
9	.1	1	
10	.01	8	
11	.01	4	
12	.01	2	
13	.01	1	
14	.001	8	
15	.001	4	
16	.001	2	LSB

MSB = Most Significant Bit
LSB = Least Significant Bit

Pin Connections to the HP Digital Input Port

Figure 5

transducers over a range of 0.002 - 19.998 PSI. The scope of possible pressure values to be encountered in this thesis project are well within this measurement range.

E. APPLICATION SOFTWARE

The BASICA application program that runs the hotwire experiment is composed of three parts. The first is called a STATEFILE. The Statefile is a program automatically compiled by the HP Soft Front Panels software. It tells the computer what configuration each unit was left in when the instruments were last used (i.e., Relay Multiplexer was set for channel one as the input and the output device was enabled, etc.). The Statefile in use in the hotwire laboratory is named HOTWIRE.HPC and is located in the hotwire sub-directory on the computer's fixed C-disk.

The second part of the BASICA application program is called the PROGRAM SHELL. The Program Shell consists of BASICA program lines that perform initialization chores to allow communication between the HP instruments and the IBM computer. In essence, the Program Shell lets the computer know every function available at each of the data acquisition instruments. When one of these functions is called up in the BASICA application program the computer already "knows" that function exists and where to find it. The Program Shell, like the Statefile, is created by the HP system Soft Front Panel software.

The third part of the BASICA application program is the actual application code. The Program Shell occupies lines 1 through 999. The application code starts at line 1000 and begins to run only after all system initialization is complete. The application code consists of a combination of PC instruments statements with BASICA keywords and programming statements. The hotwire laboratory is run in two parts using BASICA programs "CALIB" and "PART2". These programs are discussed in detail in Chapters 4 and 5, respectively.

Certain limitations exist when using the BASICA language to run a long, complicated data acquisition program. A BASICA application program, including the Program Shell must be less than 64 kilobytes in length. This is a limitation imposed by BASICA version 2.0 in use by the IBM Disk Operating System. Since the application code contains all screen graphics, PC instrument statements, plus data manipulation and storage, the code can easily exceed this 64 Kilobyte limitation. For this reason the hotwire laboratory experiment exists in two parts.

Transfer of control from CALIB to PART2 is accomplished automatically through a BASICA "LOAD" and "RUN" command. When LOAD PART2 is executed the BASICA program CALIB is removed from the allotted working memory and PART2 is put in its place. The RUN command automatically causes the experiment program "PART2" to run without any further user input. When employed in this manner, the application codes can be as long and complicated as necessary and never exceed the memory allocation of 64 Kilobytes.

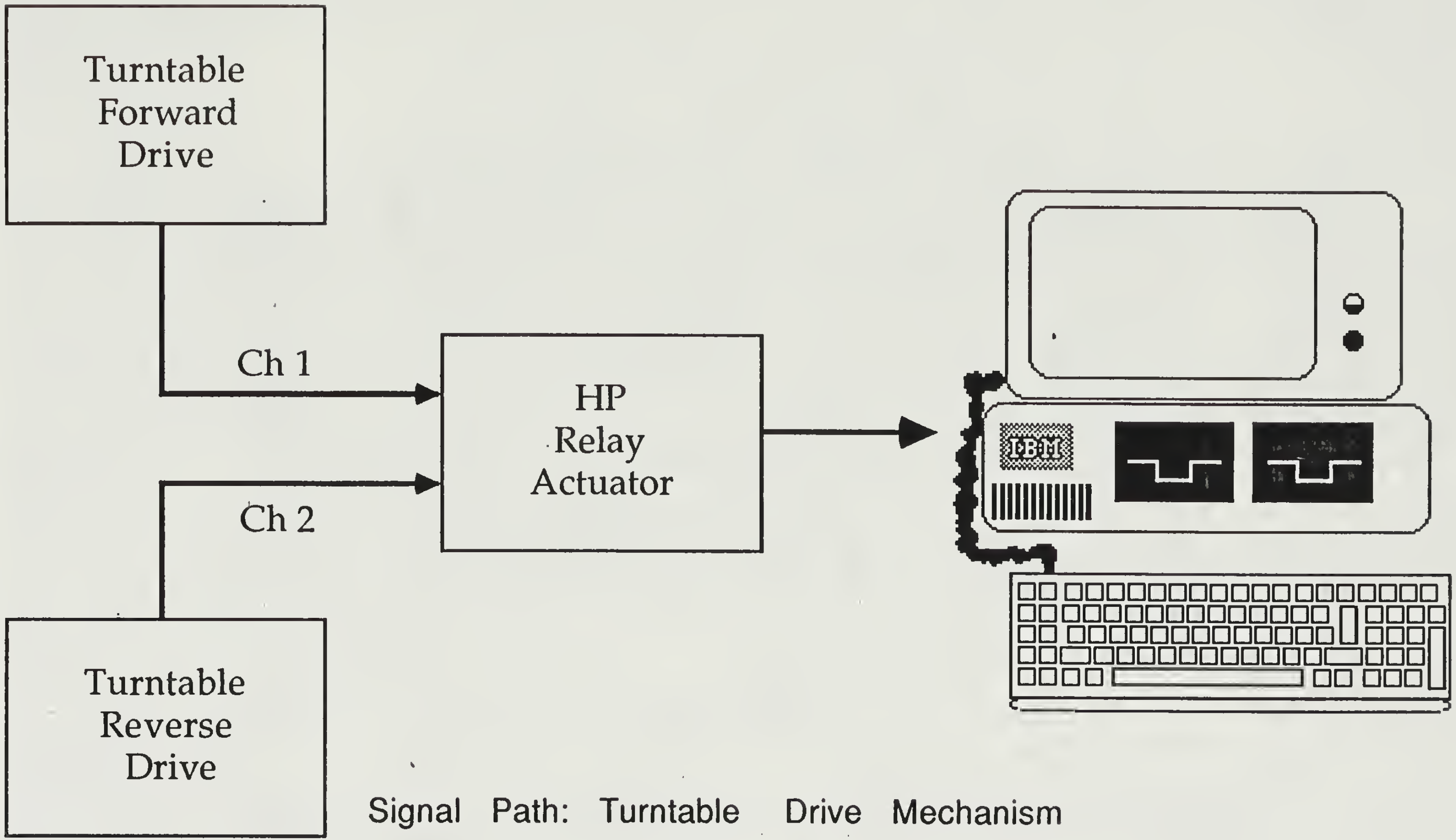
Seven named variables are transferred from CALIB to PART2. Four relate the calibrated range of wind tunnel velocities for subsequent use by the hotwire experiment. Two variables contain the calibration curve slope and intercept data. The seventh variable contains the calibration tunnel temperature in degrees Farenheit. This is used for hotwire compensation when the tunnel temperature varies during the experiment. The seven variables are written to a permanent data file that is updated with each experiment. The updated data file is read into PART2 after the RUN command has been executed.

III. DATA ACQUISITION

A. ANGLE OF INCIDENCE

A ten turn potentiometer is located on the drive mechanism of the wind tunnel turntable. Potentiometer data is routed to signal amplifier number two in the instrumentation rack. Signal amplifier output is then directed to channel five of the Relay Multiplexer. Voltage output from the signal amplifier is zeroed at 0 degrees AOA and is positive for forward turntable motion and negative for reverse motion. Voltage readings are scaled and converted to degrees for angle of incidence measurements in the BASICA application code of CALIB and PART2.

Turntable control for forward motion (positive AOA) is routed to channel one of the Relay Actuator. Turntable control for reverse motion (negative AOA) is routed to channel two of the Relay Actuator. Figure 6 shows signal path channels leading to turntable control. When forward turntable motion is desired, Relay Actuator channel one contacts are closed. Timing for Relay Actuator closure and turntable motion is scaled to account for the diminishing difference between the AOA desired and the present AOA. When the turntable nears the desired AOA the relay contacts will close for only a portion of a second to allow for an accuracy of +/- 0.1 degree. After each Relay Actuator



Signal Path: Turntable Drive Mechanism
Figure 6

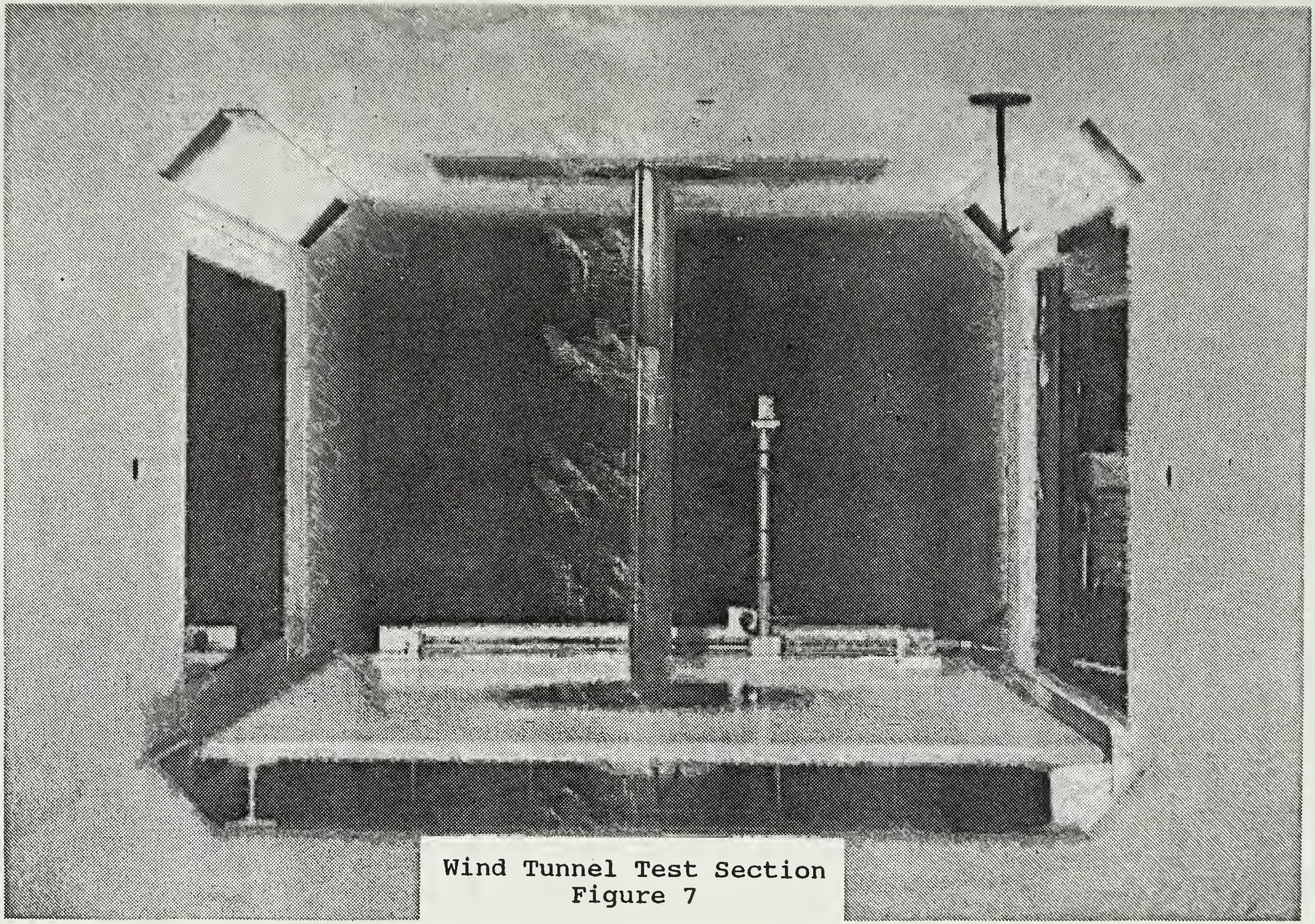
closure (resulting in turntable motion) the present AOA is read and the new difference between the desired AOA and present AOA is computed. This sequence is identical for reverse turntable motion except Relay Actuator channel two contacts close.

B. TRAVERSE ASSEMBLY POSITION

A potentiometer on the traverse assembly sends its output to signal conditioner number three, located in the top of the instrumentation rack. The potentiometer rides a gear train across the wind tunnel test section. Potentiometer output is zeroed at the near end of the test section and increases linearly to the far side. A DC voltage signal from signal conditioner number three is routed to input channel four of the Relay Multiplexer. Scaling of the input voltage to "inches" takes place in the BASICA application code of PART2. Figure 7 shows the traverse assembly mounted in the wind tunnel test section behind the airfoil model.

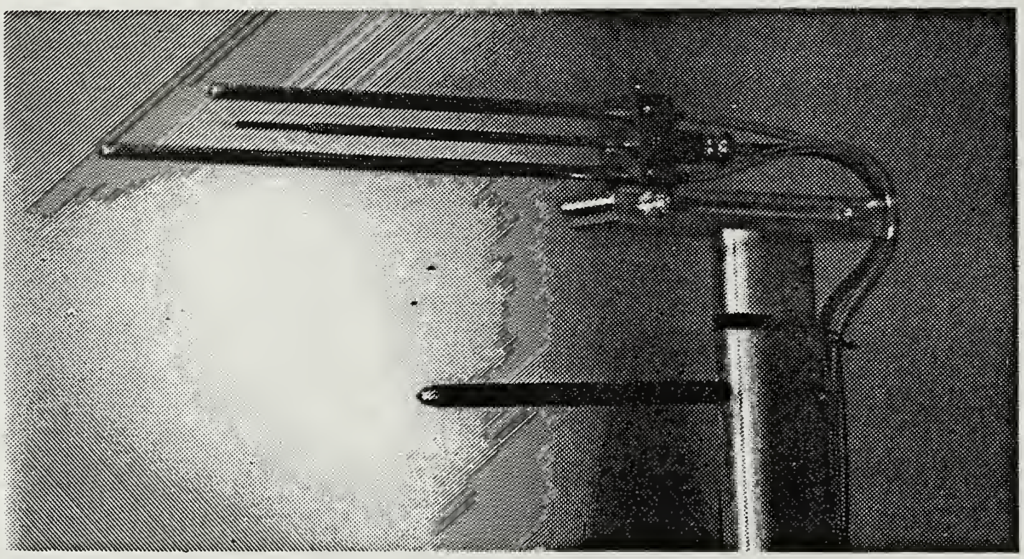
C. PITOT-STATIC PRESSURE

The upper sensor on the traverse assembly in Figure 8 senses wind tunnel total pressure. This pressure makes up one half of a pitot-static reading for tunnel flow field measurement and is connected to one side of a differential



Wind Tunnel Test Section
Figure 7

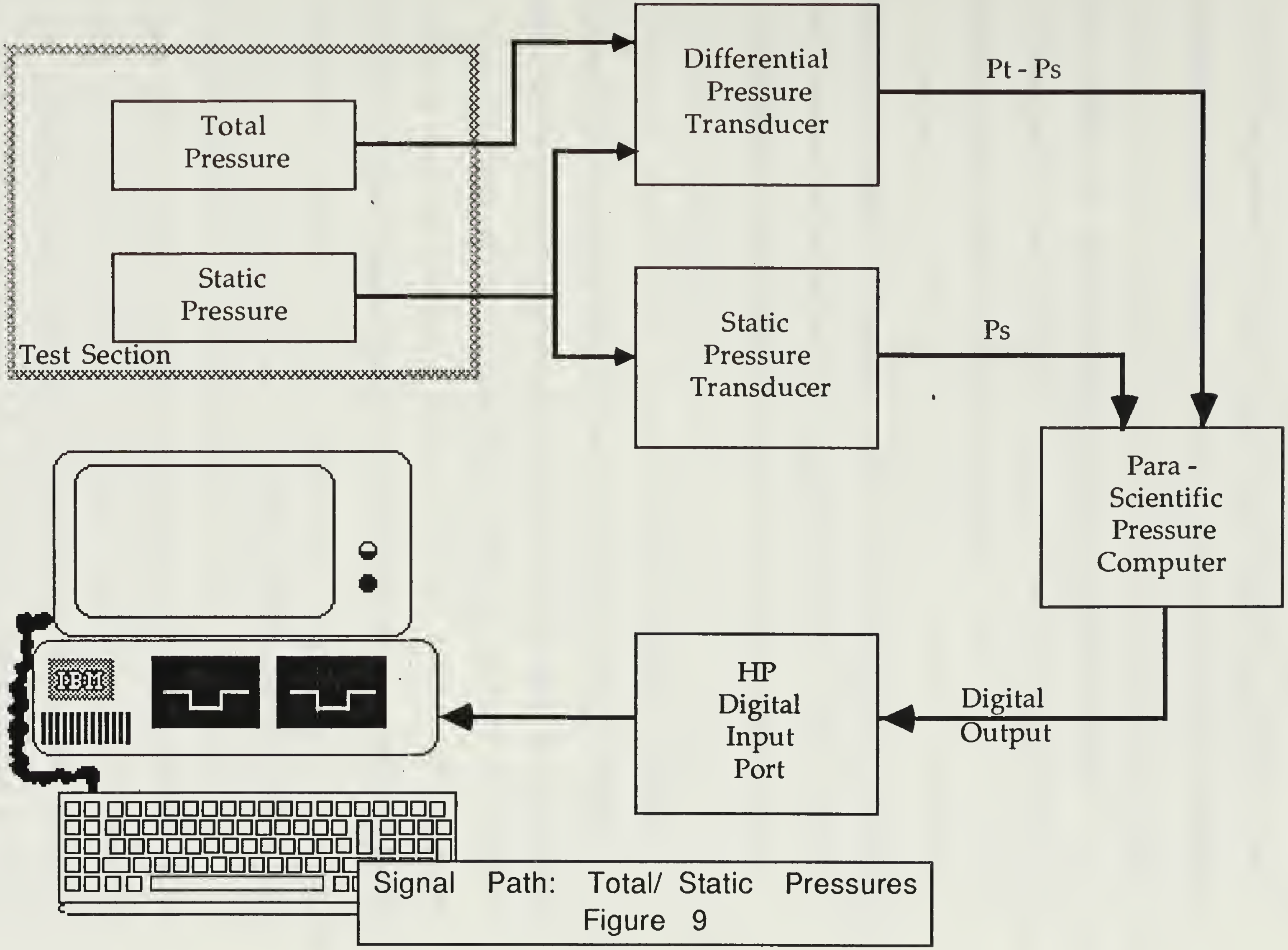
pressure transducer. The longest sensor on the traverse unit (third from the top in Figure 8) senses static pressure. This reading makes up the second half of the pitot-static measurement and is routed to the other side of the differential pressure transducer. Pressure data from the differential transducer is routed to input P1 on the Para-Scientific Pressure Computer.



Pitot-Static/Hotwire Traverse assembly

Figure 8

Although the two sensors occupy separate tubes, they function exactly as a conventional pitot tube. Signal paths for this system are shown in Figure 9. The traverse assembly pitot-static system was constructed in this manner to measure static and stagnation pressures at the same axial


Total Pressure
Static Pressure
Test Section
Differential Pressure Transducer
Static Pressure Transducer
Pt - Ps
Ps
Para - Scientific Pressure Computer
Digital Output
HP Digital Input Port
IBM

Signal Path: Total/ Static Pressures
Figure 9

location. The total and static pressure sensors as well as the hotwire probe, all obtain their flow measurements on the same vertical plane aft of the airfoil. Wake profile readings taken from the hotwire or pitot-static system are virtually identical.

Static pressure in the test section is also used for air density and flow velocity calculations. The static pressure line that is routed to the differential pressure transducer is also connected to an absolute pressure transducer by means of a "T" connection. Static pressure readings from the absolute pressure transducer are connected to input P2 on the Para-Scientific Pressure Computer.

D. HOTWIRE VOLTAGE DATA

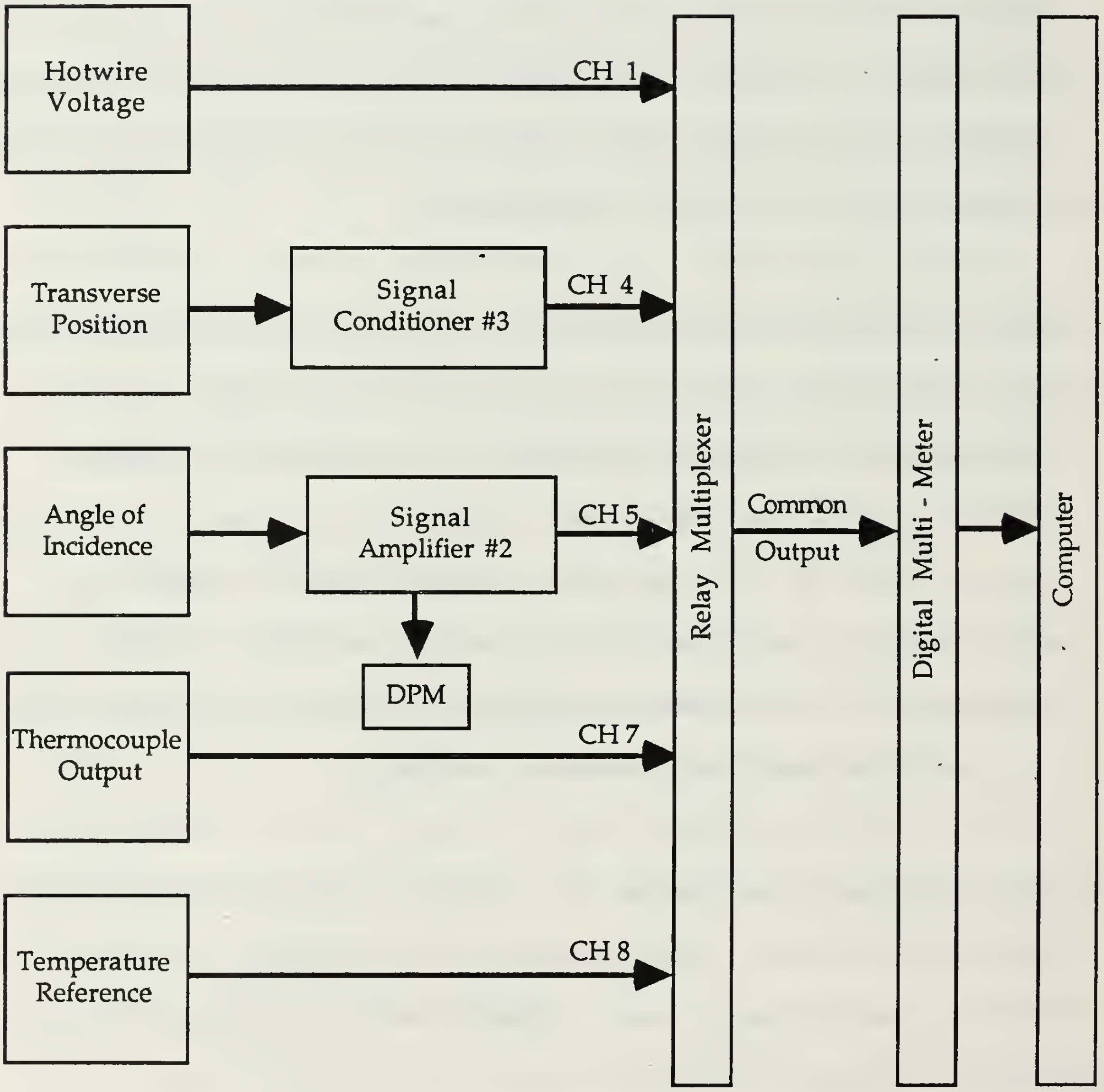
Hotwire voltage data is output directly in DC volts from the back of the IFA 100 thermal anemometer. The voltage output is routed to input channel 1 of the Relay Multiplexer. This value is read directly by the Digital Multi-Meter and at no time are the values scaled or altered.

E. WIND TUNNEL TEMPERATURE

A Copper-Constantin thermocouple is located inside the wind tunnel in the converging section prior to the test section. Thermocouple output is routed directly to input channel seven of the Relay Multiplexer. An internal sensor in the Relay Multiplexer monitors and converts ambient room temperature to a reference output voltage. This reference voltage is connected to input channel eight of the Relay Multiplexer. Voltage output from the temperature thermocouple to channel 7 and reference voltage output to channel 8 are read by the DMM consecutively and their values are added to obtain a final accurate wind tunnel temperature reading. A subroutine for measuring wind tunnel temperature is located in both CALIB and PART2.

NOTE: Coefficient values used in the temperature measurement subroutine are extremely dependent on the type of temperature thermocouple used. If the present Copper-Constantin thermocouples are replaced or altered in future research the HP PC Instrument User's Manual should be consulted for information on new thermocouple coefficients. Any change to the present system can result in spurious or inaccurate tunnel temperature readings.

NOTE: Figure 10 contains all signal paths utilizing the HP Relay Multiplexer. Data acquisition channels not using the HP Relay Multiplexer were illustrated in Figures 6 and 9.



Signal Paths

Figure 10

IV. HOTWIRE CALIBRATION

Heat loss from the hotwire sensor due to forced convection is dependent on many parameters:

flow velocity - magnitude and direction
probe geometry and material
temperature of the sensor

Although some of these influences can be kept constant for a particular experiment no sufficiently exact theoretical relationship for thermal anemometer heat transfer exists. Thus, the need for system calibration.

For a given probe in the low speed wind tunnel the wire response E, for constant temperature anemometry is a function of:

flow velocity
yaw angle
pitch angle

or E = f(velocity, yaw angle, pitch angle). The yaw and pitch angle remain zero in this thesis as the traverse assembly is not built to swivel. The probe always faces directly into the flow as the traverse unit moves across the test section. The aim of hotwire system calibration is to determine the function f in analytic or tabular form.

Hotwire system calibration is accomplished through the first BASICA program called CALIB. To calibrate a hotwire sensor, a comparison is made of the hotwire voltage output to a known value of flow velocity in the wind tunnel test section. Air flow in the NPS wind tunnel can be considered low speed, steady, incompressible flow. Under these conditions a pitot tube will give extremely accurate flow velocity measurements. Thus, the pitot tube becomes the standard by which hotwire probe output is compared.

The calibration curve is created by running the wind tunnel at the lowest velocity to be used in the experiment and incrementally increasing the tunnel velocity. At each discrete wind speed, hotwire output and tunnel velocity values are sampled. The selected tunnel speeds range from the lowest to highest velocity to be encountered in the experiment. Since the hotwire system's accuracy is only known within the calibration range, all expected experimental velocities are covered. CALIB uses a least squares analysis to determine the best linear fit through the calibration data points. Figure 11 is representative of all calibration curves using this method. Deviation from the linear fit for each data point is typically very small.

One suggestion for future experiments at NPS is to run the wind tunnel during the data collection phase of an experiment in the middle of the calibrated range of velocities. If a series of velocities are to be used, care should be taken to ensure that they are within the calibration range. A calibration curve can be constructed at

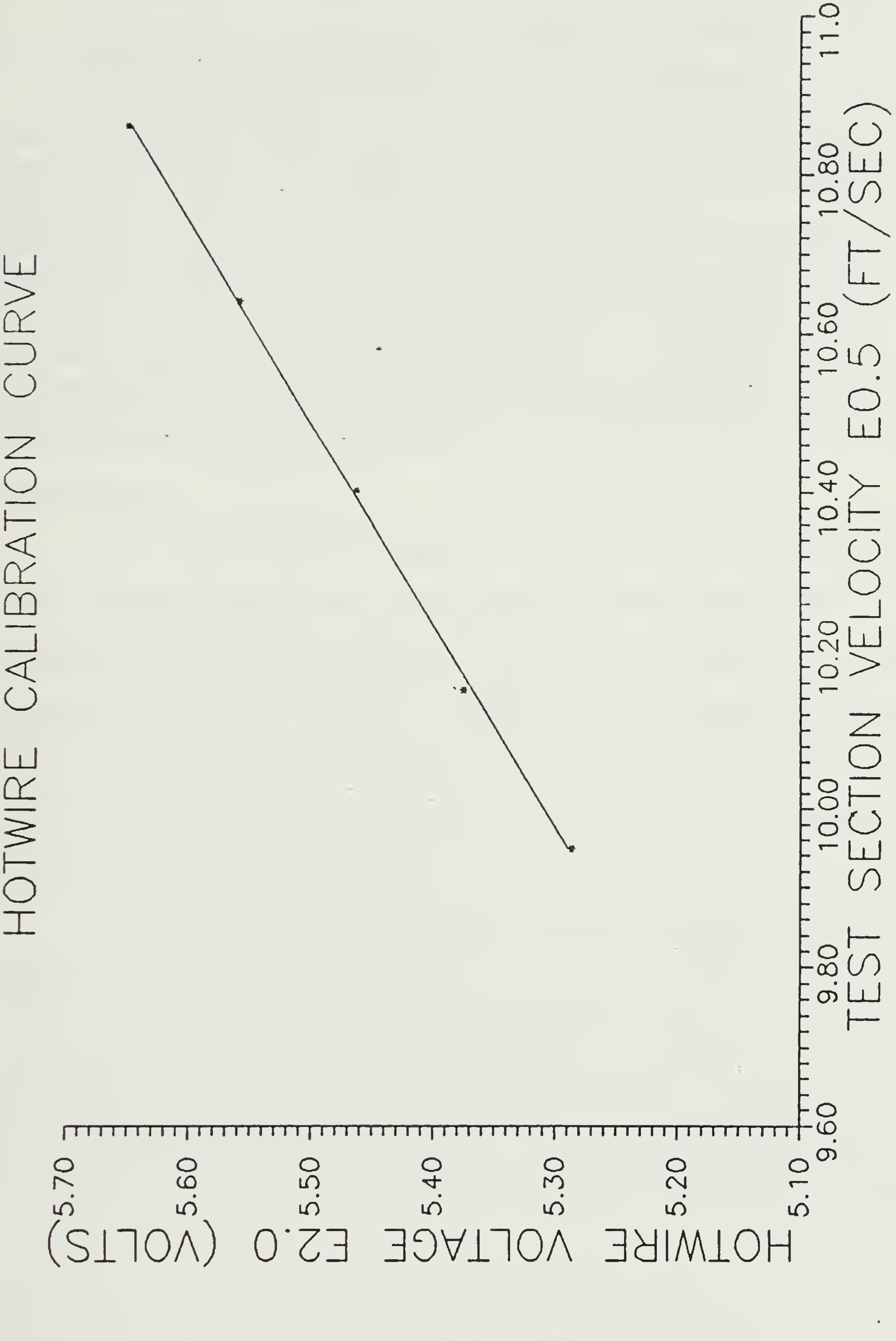

HOTWIRE CALIBRATION CURVE
HOTWIRE VOLTAGE E2.0 (VOLTS)
5.10
5.20
5.30
5.40
5.50
5.60
5.70
TEST SECTION VELOCITY E0.5 (FT/SEC)
9.60
9.80
10.00
10.20
10.40
10.60
10.80
11.00

Figure 11

any time. If the validity of experimental data is in question it is best to go back to the begining and re-create the calibration curve.

A minimum of 10 data points should be taken to construct the calibration curve. The accuracy of the curve is directly related to the number of inputs. For time consideration in classroom demonstrations, 10 points will prepare a curve with accuracy to 1-2%. If the hotwire system is to be used for research or other important measurements, 20-30 data points are necessary to achieve optimum accuracy.

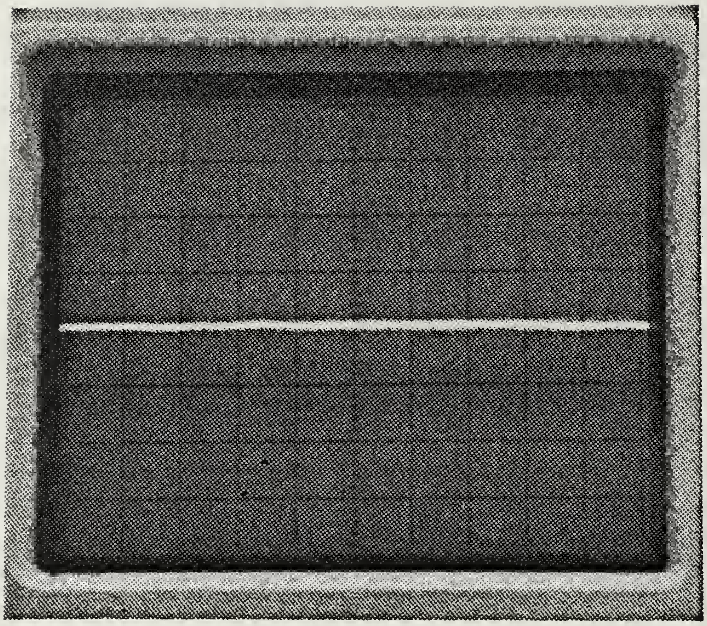
V. DATA ANALYSIS

A. EXPERIMENTAL TECHNIQUE

The experimental portion of this thesis is performed by the BASICA application code called PART2. PART2 starts by prompting the user to input desired angles of attack to be analyzed. When the first angle of attack is set in the test section a flow field velocity update is performed. If adjustments are necessary the user can re-adjust test section wind speed and update the tunnel velocity repeatedly.

After the desired angle of incidence and test section flow velocity are set, the system is ready for data sampling. Traverse position in the test section is initially displayed on the computer screen. The researcher can move the traverse assembly anywhere across the test section for the first data point. Accuracy of the traverse position is +/- 0.01 inch.

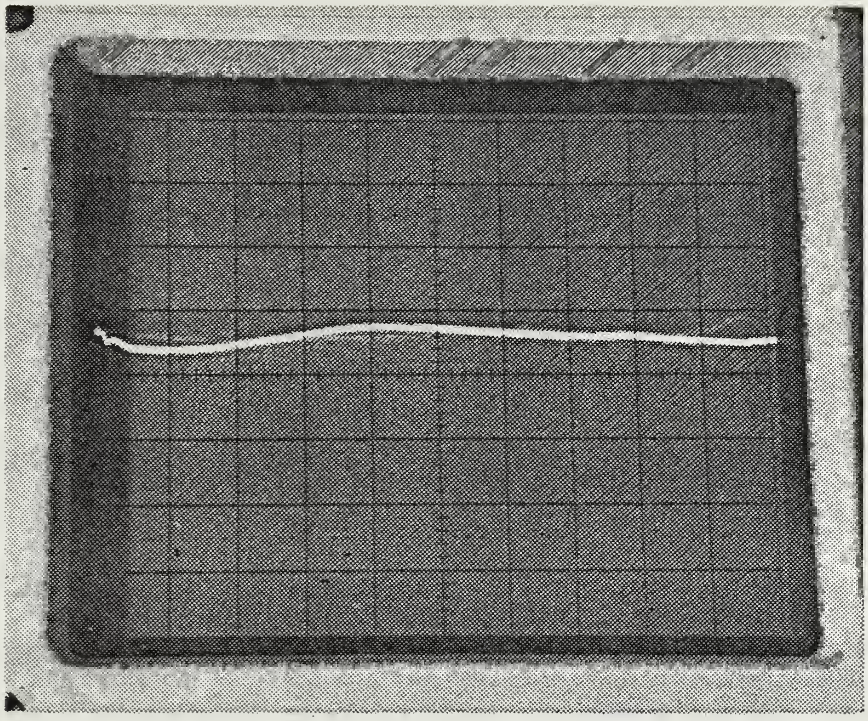
The airfoil wake velocity profile is the focus of interest in this experiment. In order to concentrate on the wake profile and not spend excessive time sampling the free stream, correlation with the oscilloscope is necessary. When the hotwire system is running, its electrical response to the test section flow field is output to the oscilloscope.



Oscilloscope Response To Steady Flow

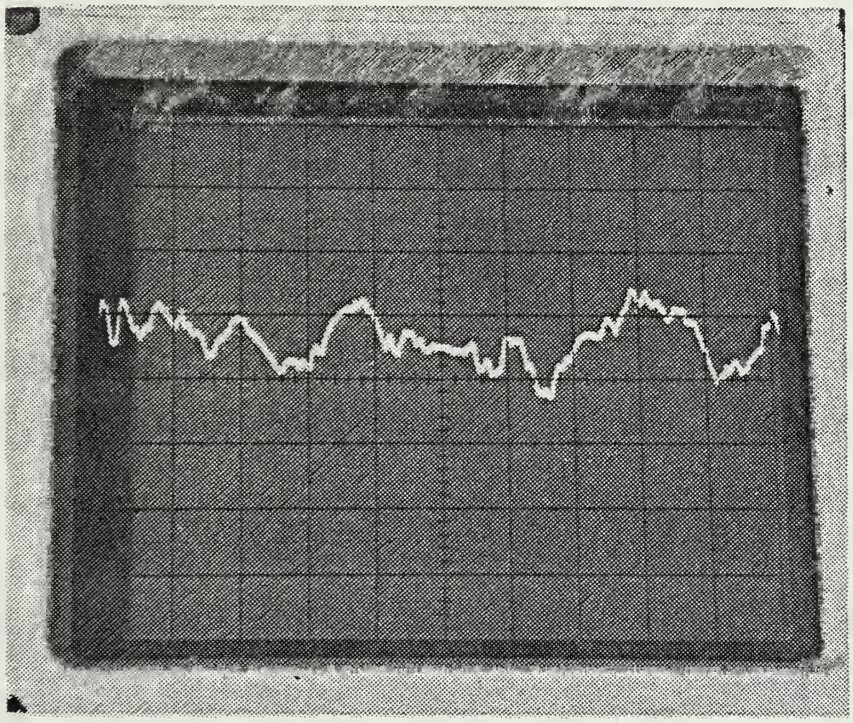
Figure 12

The oscilloscope trace will be stable under steady flow conditions (see Figure 12). As the hotwire sensor nears the airfoil wake the oscilloscope trace will begin to jump in response to the disturbed air flow at the wake boundary (see Figure 13). When the hotwire sensor is in the fully developed wake the oscilloscope will display the unsteady response caused by turbulent flow (see Figure 14). By correlating the hotwire sensor response as seen on the oscilloscope with the traverse position as given on the computer screen, the start and end of the wake boundary can easily be detected.



Oscilloscope Response To Wake Boundary

Figure 13



Oscilloscope Response To Turbulent Flow

Figure 14

Experimental procedure consisted of moving the traverse assembly toward the wake until an unsteady response was displayed on the oscilloscope. Once the position of the wake edge is known, the traverse unit is backed out of the wake into the freestream and measurements are initiated. By correlating traverse unit position with hotwire sensor response, experimental data sampling can be accomplished in a more efficient manner.

When the traverse assembly is positioned for a reading, the computer samples the hotwire system voltage and pitot-static pressures for independent test section velocity measurements. This procedure is repeated as many times as desired to obtain a full wake velocity profile of the airfoil. If PART2 is to be used as a classroom demonstration as few as three data points per inch of traverse travel across the test section will provide a viable velocity profile graphical output. If the experiment is to include a drag coefficient measurement of the airfoil, at least five data points per inch of traverse travel is recommended. Ten points per inch is optimal.

B. DRAG COEFFICIENT MEASUREMENT

PART2 is capable of measuring the drag coefficient of the airfoil at each angle of incidence. The drag coefficient is deduced as a loss of linear momentum in the air flow as it passes over the airfoil. At each data sampling position the hotwire and pitot-static readings are converted into a dynamic pressure value. Dynamic pressure in the wake, q, is compared in a ratio to the dynamic pressure of the freestream, qo. The dynamic pressure ratio q/qo, is approximated by the following equation:

$$q/qo = V2/V1(1-V2/V1)$$

where V2 is the flow velocity in the wake and V1 is the freestream velocity.

This method of analyzing the wake velocity profile will only work if the traverse unit is situated far enough behind the wing so that the wake has returned to tunnel static pressure. In early experimental trials of this thesis the static pressure profile was measured across the wake for all angles of incidence. In every case the change in static pressure across the test section was zero. Thus, obtaining profile drag coefficients by the momentum method in the NPS wind tunnel is fully justified (see Chapter VI). Stalled conditions were not included in this analysis as the momentum method breaks down. A large part of the drag in this case is caused by rotational losses and does not appear as a drop in linear momentum.

The dynamic pressure ratio, q/qo, is calculated for each data point in the wake velocity profile. If the dynamic pressure ratio were to be plotted on an X-Y axis (test section position on the X axis, q/qo on the Y axis) it would look like an inverted velocity profile. The area under this "curve", multiplied by (2/airfoil chord) makes up an uncorrected value for the profile drag coefficient. The term uncorrected refers to the fact that the drag coefficient is based on a dynamic pressure, q, which would exist if no airfoil were present in the test section. The uncorrected drag coefficient is calculated using hotwire system data as well as pitot-static system data for comparison purposes.

Integration of the dynamic pressure ratio is accomplished by a modified Trapezoidal routine. [Ref. 1: pp. 316-325] Normally trapezoidal integration is carried out by averaging a function over an incremental area, then multiplying the average by the interval length. Round off errors can become excessive if the function to be integrated varies rapidly. To reduce round off errors an Akima Spline fitting technique is used to fit a third order polynomial between each pair of data points. Trapezoidal integration is then used to sum up all incremental areas with a cubic fit between data pairs vice an average value.

C. TUNNEL CORRECTIONS FOR DRAG ANALYSIS

Testing an airfoil section in a wind tunnel is not the same as testing in a free air stream. The test section walls provide constraints to the air flow that must be corrected. The NACA 66(215)-216 airfoil tested in this thesis is a two dimensional airfoil section, therefore two dimensional corrections apply. [Ref. 2:pp. 7.1-8.3] dimensional airfoil section, therefore two dimensional corrections apply. [Ref. 2:pp. 7.1-8.3]

For the testing conditions encountered in this experiment only two corrections are relevant. The first is known as "solid blocking". With the airfoil in the wind tunnel the test section cross sectional area is reduced. A reduction in area causes the surrounding flow to speed up according to Bernoulli's law. Solid blocking results in an increase in forces and moments on the airfoil at any given angle of incidence.

The second correction is called "wake blocking". The wake behind the airfoil has a lower velocity than that of the freestream. To satisfy continuity the air flow outside the wake must speed up to maintain a constant volume of air flowing through the test section. The increase in free stream velocity near the wake also results in an increase in forces and moments on the airfoil. In this case the effect increases with an increase in wake size (i.e., at high angles of incidence).

The solid blocking and wake blocking corrections are applied to the uncorrected drag coefficients to produce a final corrected profile drag coefficient.

VI. RESULTS

A. WAKE VELOCITY PROFILES

Figures 15 through 19a represent wake velocity profiles obtained through the hotwire and pitot-static systems. In all cases the profiles are remarkably similar. The profiles highlight two main points concerning hotwire anemometry in the NPS wind tunnel. First, the hotwire anemometry system works quite well. Second, when turbulence intensity is not too high (angles of incidence less than ten degrees) the hotwire system has the greater sensitivity in unsteady flow. In Figures 15,16, and 17 the hotwire profile consistently reaches a lower peak at the wake center than the pitot-static system. The increased sensitivity falls off at higher turbulence intensities caused by large angles of incidence (see Figures 19 and 19a).

An explanation of the spurious data points or jumps in an otherwise smooth velocity profile requires a reiteration of how the data is obtained. The Digital Multi-Meter is capable of updating its voltage measurements (with an acceptable accuracy) at a rate of only 2.5 readings per second. When a measurement is called for from the application programs, the last full measurement obtained by the Digital Multi-Meter is sent to the computer. These "snap shot" measurements reflect the unsteadiness of the flow field in the wake of the airfoil.

The data acquisition system used in this thesis was designed to measure steady flow conditions. To simulate a steady state (or time averaged) data reading many consecutive "snap shot" measurements are taken by the Digital Multi-Meter and averaged to represent a mean value. In performing a wake profile analysis a trade off must be accepted between the number of measurements taken for each data point and the time allotted for data collection.

Each of the velocity profiles except Figure 19a were prepared using 70 consecutive measurements per data point. In a high turbulence intensity flow, 70 measurements per data point is clearly insufficient for a smooth graphical velocity profile output. Figure 19 demonstrates that only a rough approximation of the actual velocity profile can be made in high turbulence regimes. The hotwire velocity profile of Figure 19a was prepared using 300 measurements per data point for comparitive purposes. The pitot-static profile remained at 70 readings per data point. The hotwire curve is much smoother in this case, proving that a time averaged value of the wake velocity will provide the smoothest plot. It is also interesting to note that the wake profiles measured in Figure 19 were completed in approximately 35 minutes. The wake profiles prepared for

Figure 19a required over 90 minutes for completion due to the extra measurements in the hotwire velocity profile.

The data acquisition system utilizing the HP Digital Multi-Meter has proven to be a useful and effective system in steady flow and low turbulence intensity regimes. Outside of these flow regimes the system loses its effectiveness and performance becomes marginal. The user should be well aware of the limitations on the system. Until another voltage measuring device capable of time averaging can be incorporated into the data acquisition system, wake profiles of 10 degrees angle of incidence and above should be viewed as little more than approximations. Highly accurate and repeatable profiles can still be obtained in low turbulence intensity flow; i.e., angles of incidence between -9 and +9 degrees.

B. DRAG COEFFICIENT ANALYSIS

Figure 20 displays corrected drag coefficients calculated from the hotwire system data and pitot-static system data. Both plots follow the same general shape, exhibiting drag buckets and steep rises in Cd at high lift coefficients. It should be noted that the drag coefficients for both systems are closer in value at lower lift coefficients. This is in keeping with the discussion on system limitations in part A. The wake velocity profiles are

are generally more accurate at lower angles of incidence, creating more accurate and repeatable drag coefficients.

The NACA drag coefficient data was obtained using the same wake survey method as used in this thesis. The difference between NPS wind tunnel derived drag coefficients and the NACA data can be explained in several ways. The NACA data on the 66(215)-216 airfoil was collected at three discrete Reynolds number flow velocities, the slowest being Re = 3.0xE6. Data was collected in the NPS wind tunnel at a Reynolds number of approximately Re = 6.0xE5. The slow tunnel velocity was primarily chosen to preserve the hotwire sensor for multiple data collection.

The airfoil model used to provide the NACA data had a 2 foot chord and spanned a 3 foot wide test section. The airfoil model used in the NPS wind tunnel has a 1 foot chord and is 32 inches in length. NACA is careful to point out that changes in surface roughness, Reynolds number, and model geometry will affect final drag coefficient calculations. Therefore an exact comparison of the NACA drag coefficient data with NPS wind tunnel derived data is unproductive. A comparison of trends is better suited for the two differing test facilities. One can conclude that the wake survey method used in this thesis will provide a very accurate value of the drag coefficient for the NPS 66(215)-216 model airfoil.

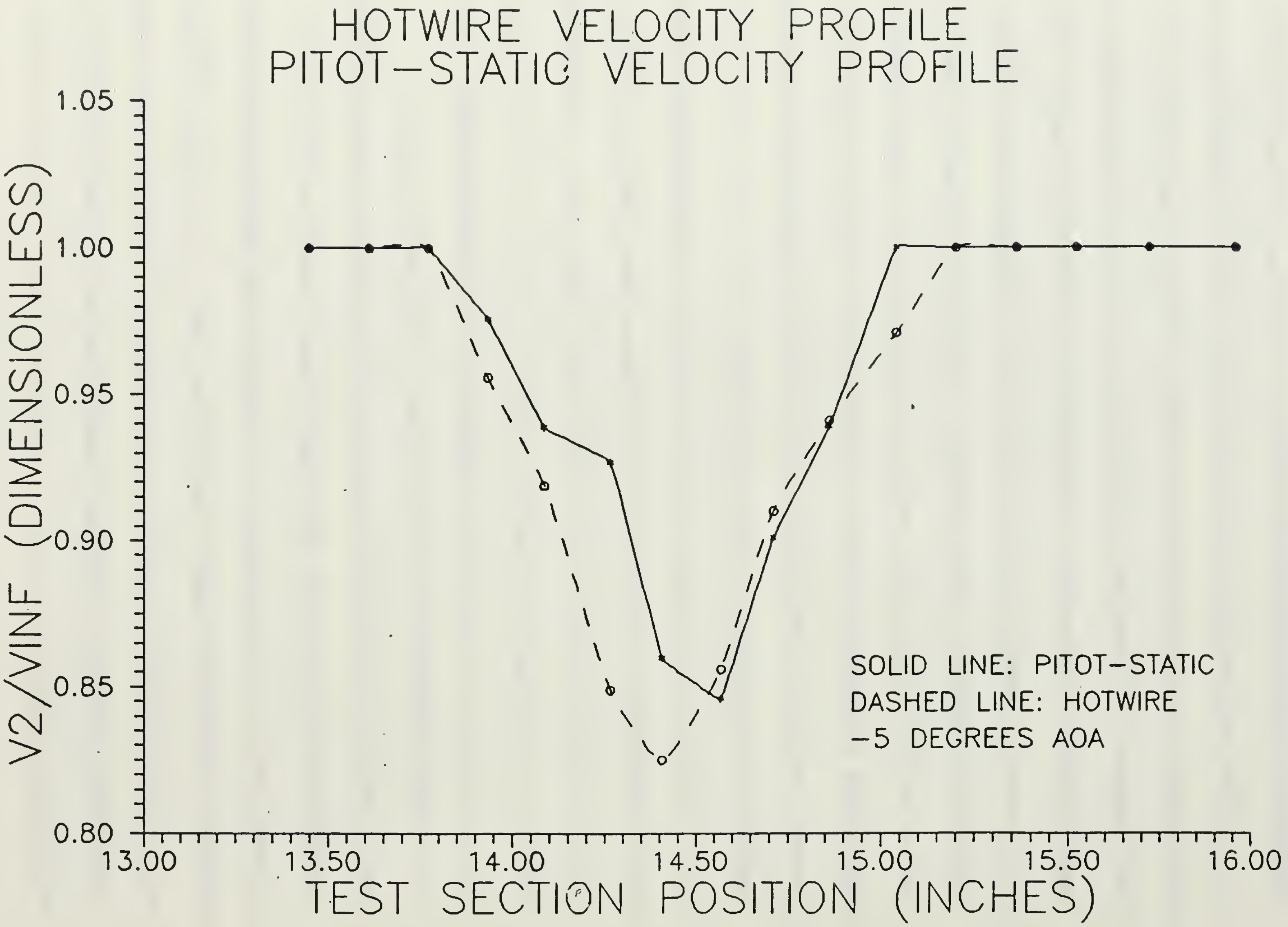

HOTWIRE VELOCITY PROFILE
PITOT-STATIC VELOCITY PROFILE
V2/VINF (DIMENSIONLESS)
1.05
1.00
0.95
0.90
0.85
0.80
13.00
13.50
14.00
14.50
15.00
15.50
16.00
TEST SECTION POSITION (INCHES)
SOLID LINE: PITOT-STATIC
DASHED LINE: HOTWIRE
-5 DEGREES AOA

Figure 15

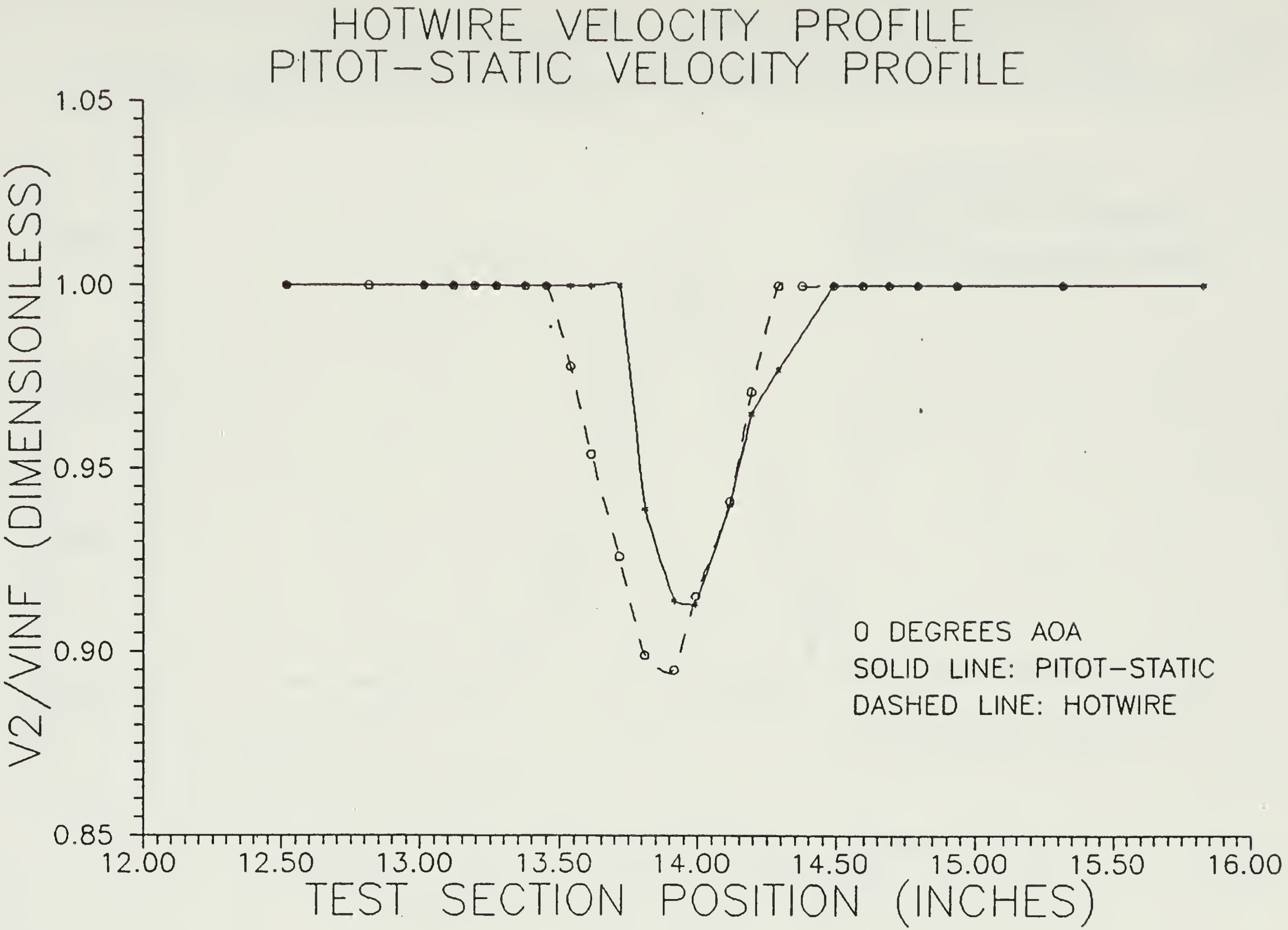

HOTWIRE VELOCITY PROFILE
PITOT–STATIC VELOCITY PROFILE
V2/VINF (DIMENSIONLESS)
1.05
1.00
0.95
0.90
0.85
12.00
12.50
13.00
13.50
14.00
14.50
15.00
15.50
16.00
TEST SECTION POSITION (INCHES)
0 DEGREES AOA
SOLID LINE: PITOT–STATIC
DASHED LINE: HOTWIRE

Figure 16

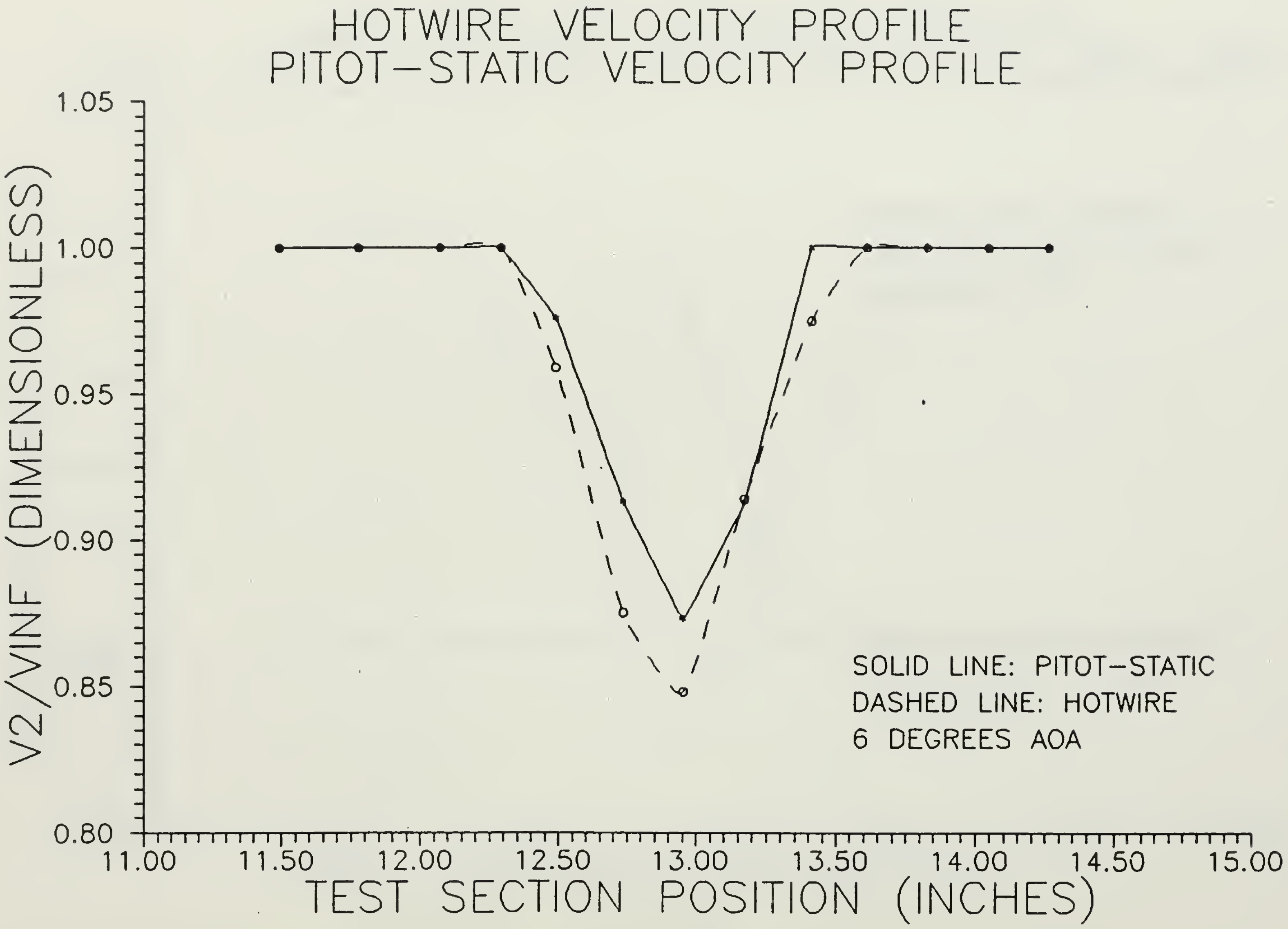

HOTWIRE VELOCITY PROFILE
PITOT-STATIC VELOCITY PROFILE
1.05
1.00
0.95
0.90
0.85
0.80
V2/VINF (DIMENSIONLESS)
11.00
11.50
12.00
12.50
13.00
13.50
14.00
14.50
15.00
TEST SECTION POSITION (INCHES)
SOLID LINE: PITOT-STATIC
DASHED LINE: HOTWIRE
6 DEGREES AOA

Figure 17

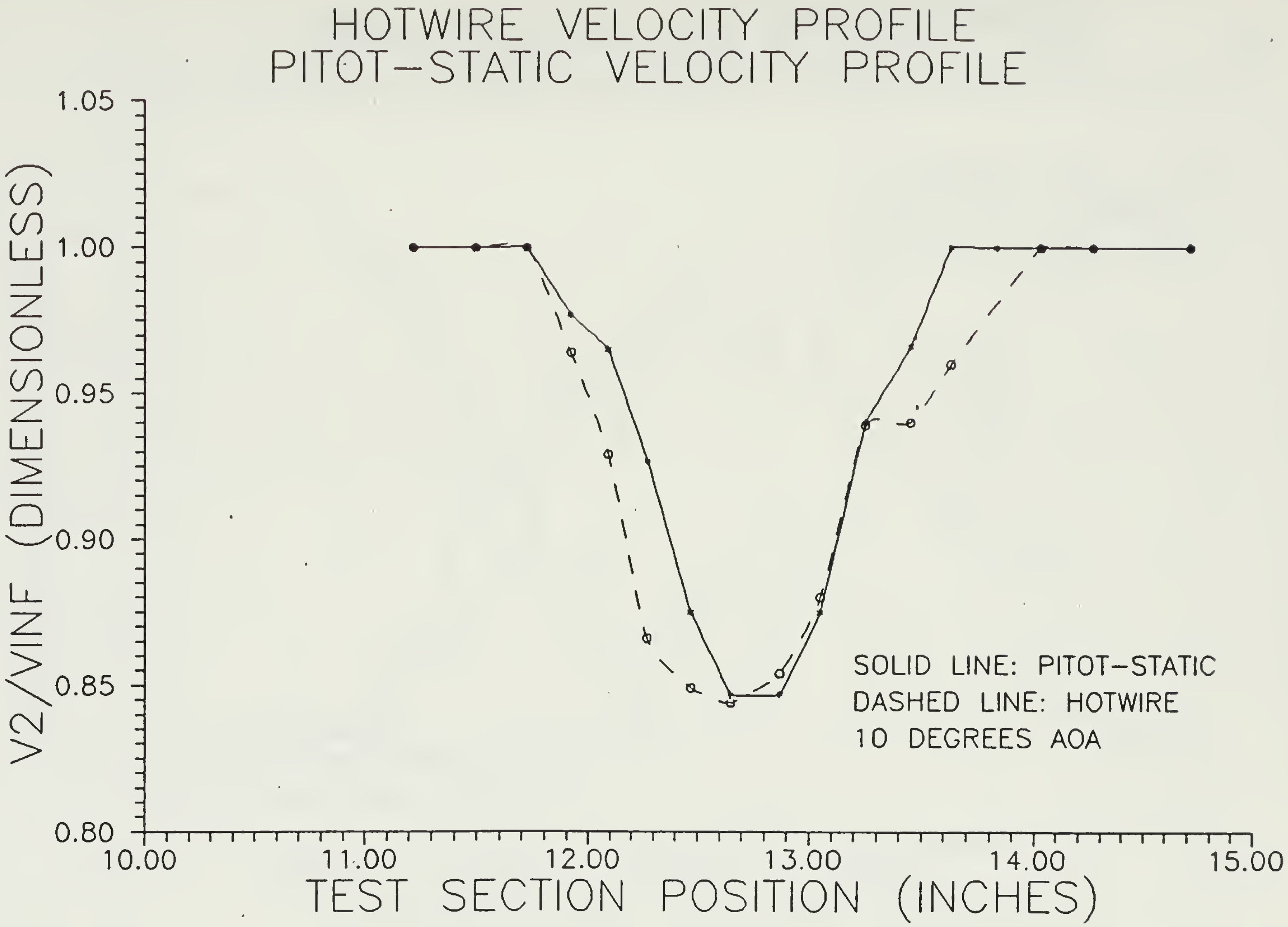

HOTWIRE VELOCITY PROFILE
PITOT-STATIC VELOCITY PROFILE
1.05
1.00
0.95
0.90
0.85
0.80
V2/VINF (DIMENSIONLESS)
10.00
11.00
12.00
13.00
14.00
15.00
TEST SECTION POSITION (INCHES)
SOLID LINE: PITOT-STATIC
DASHED LINE: HOTWIRE
10 DEGREES AOA

Figure 18

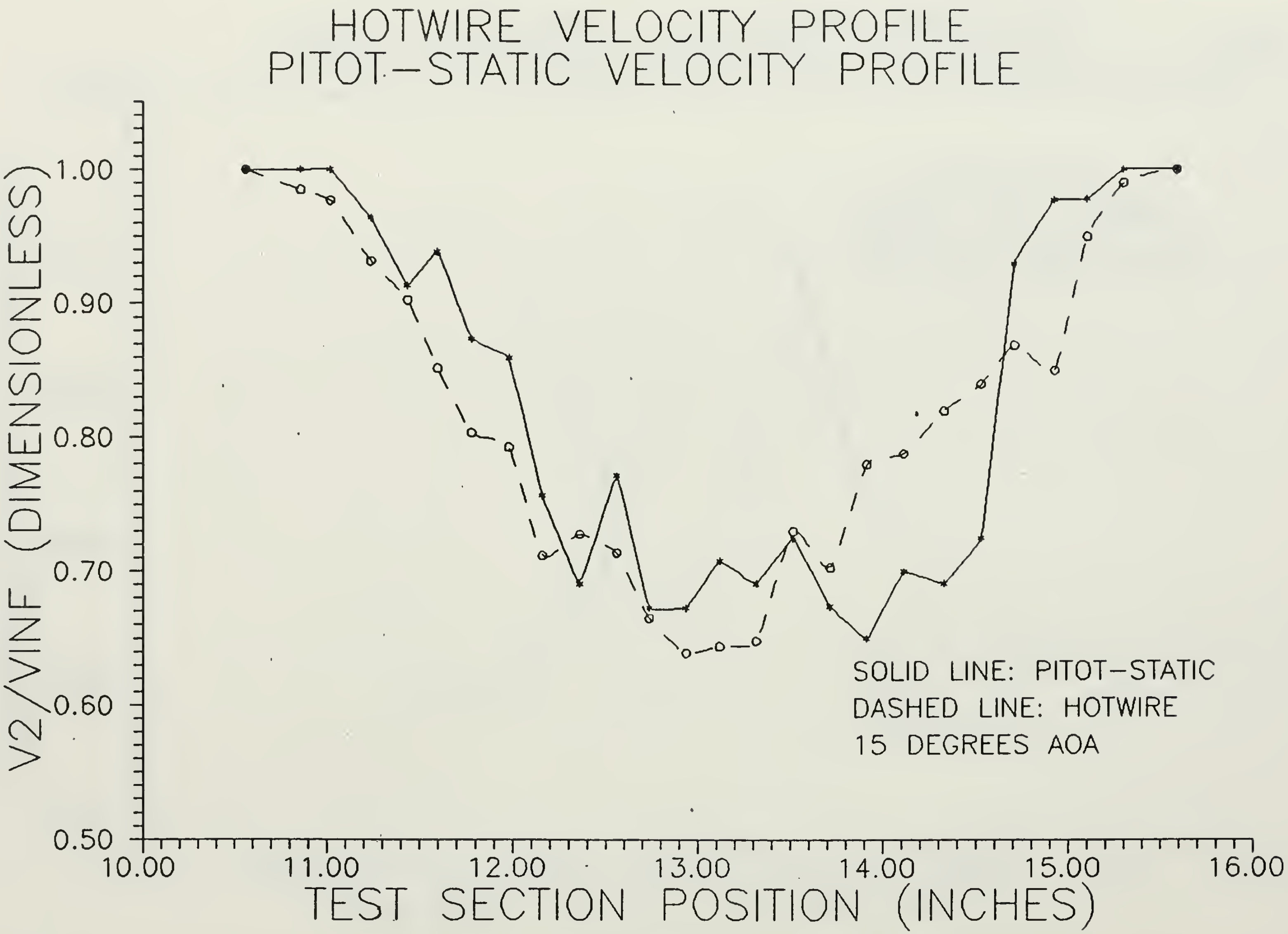

HOTWIRE VELOCITY PROFILE
PITOT-STATIC VELOCITY PROFILE
V2/VINF (DIMENSIONLESS)
1.00
0.90
0.80
0.70
0.60
0.50
10.00
11.00
12.00
13.00
14.00
15.00
16.00
TEST SECTION POSITION (INCHES)
SOLID LINE: PITOT-STATIC
DASHED LINE: HOTWIRE
15 DEGREES AOA

Figure 19

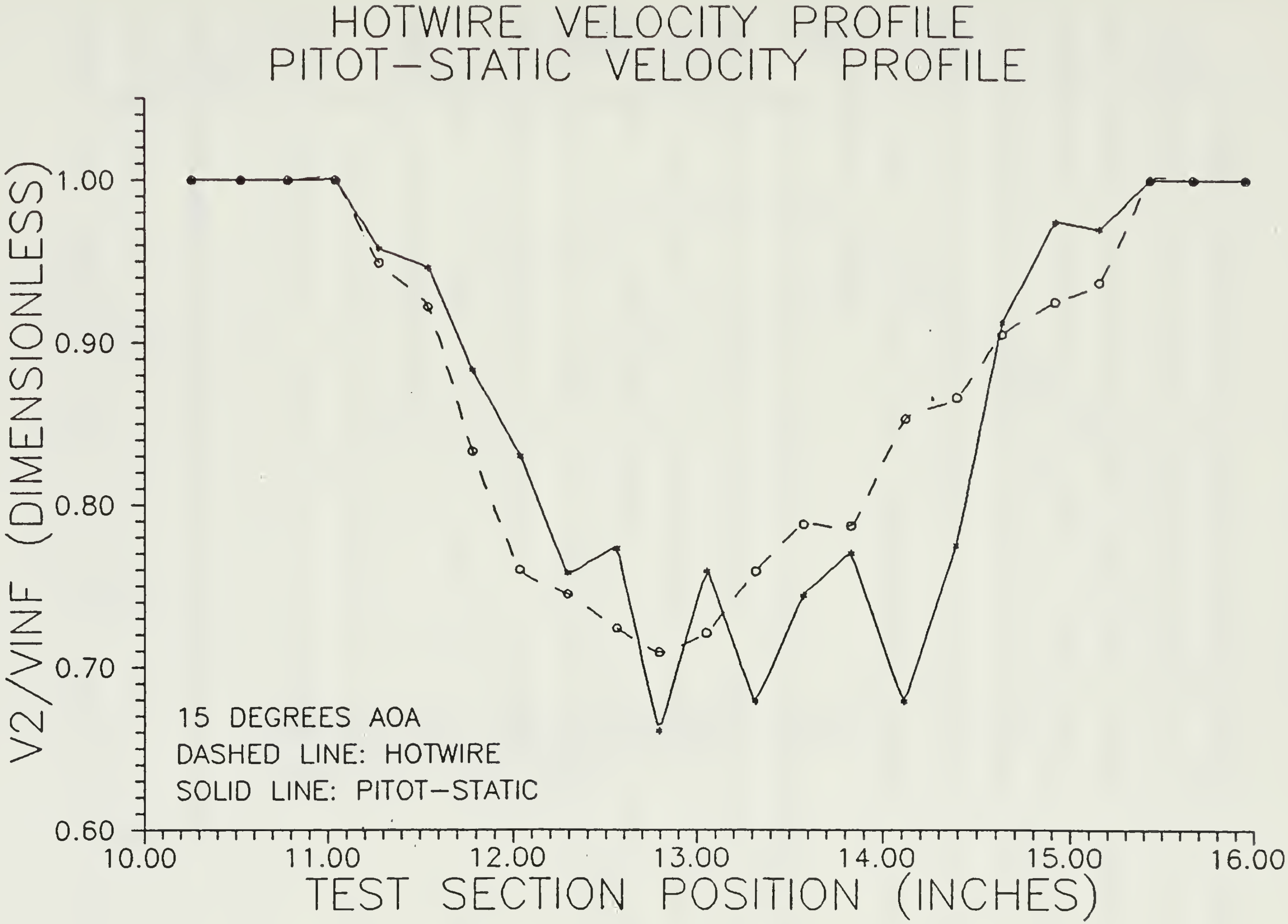

HOTWIRE VELOCITY PROFILE
PITOT-STATIC VELOCITY PROFILE
V2/VINF (DIMENSIONLESS)
1.00
0.90
0.80
0.70
0.60
10.00
11.00
12.00
13.00
14.00
15.00
16.00
TEST SECTION POSITION (INCHES)
15 DEGREES AOA
DASHED LINE: HOTWIRE
SOLID LINE: PITOT-STATIC

Figure 19a

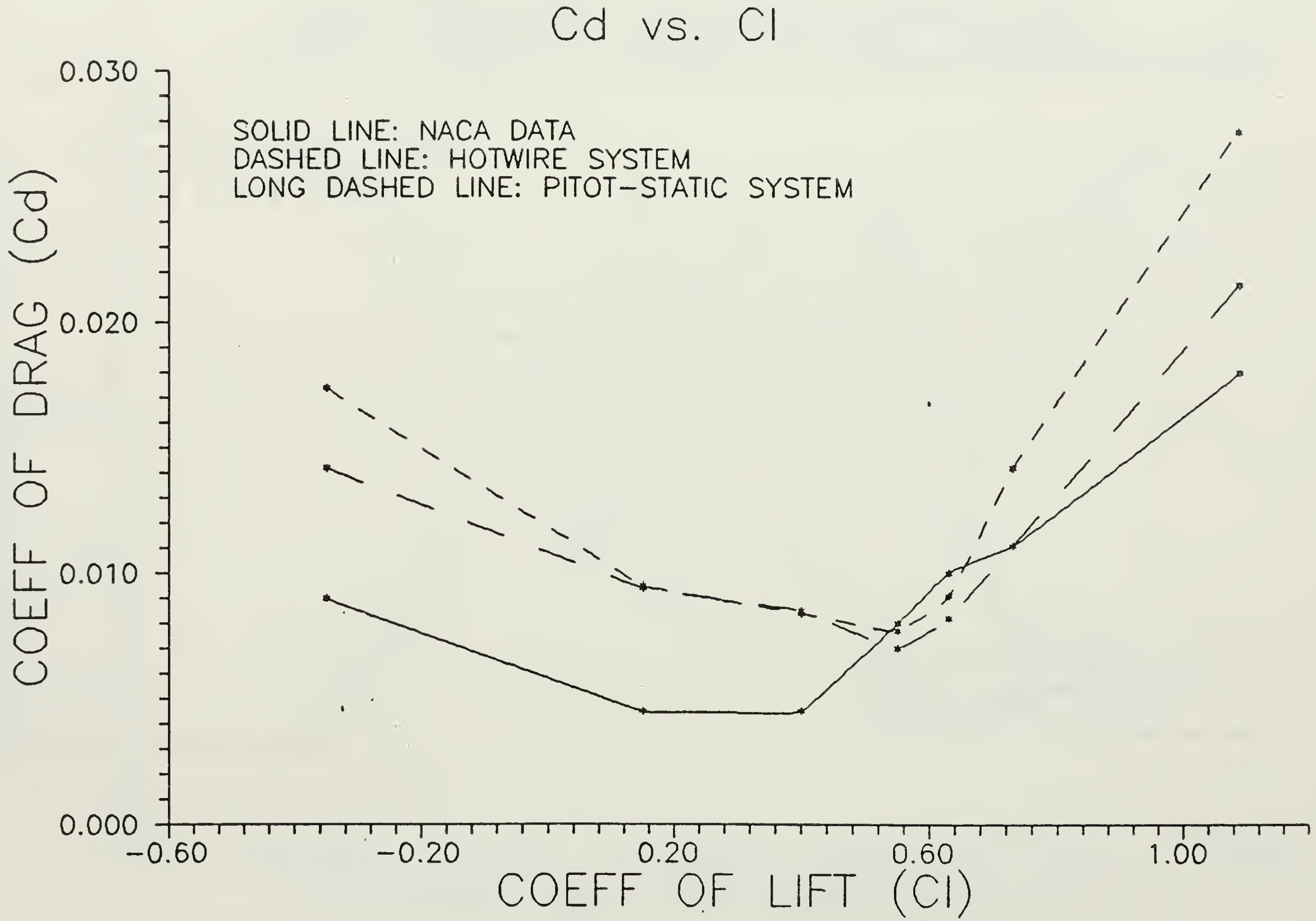

Cd vs. Cl
SOLID LINE: NACA DATA
DASHED LINE: HOTWIRE SYSTEM
LONG DASHED LINE: PITOT-STATIC SYSTEM
COEFF OF DRAG (Cd)
0.030
0.020
0.010
0.000
COEFF OF LIFT (Cl)
-0.60
-0.20
0.20
0.60
1.00

Figure 20

VII. CONCLUDING REMARKS

A data acquisition system has been installed in the NPS wind tunnel that utilizes hotwire anemometry to analyze the wake velocity profiles of two dimensional airfoil models. A conventional pitot-static system is also incorporated in the data acquisition process for comparative purposes. The hotwire data acquisition system is self contained and needs no interface with any other experiment currently in use in the NPS wind tunnel.

The hotwire data acquisition system is designed in two separate parts to enhance versatility. System calibration can be performed as a classroom demonstration or as the prelude to an entire wake profile analysis. The wake profile analysis can also be used as a classroom demonstration or as a complete experimental laboratory. In either case, all graphical and tabular data are available for online computer display or hardcopy output.

A NACA 66(215)-216 airfoil model was analyzed to verify system performance. Virtually any two dimensional wing section that will fit in the wind tunnel test section can be used for research. Wake velocity profiles and drag coefficients can also be found for non-airfoil shaped models. The lessons learned from this thesis project will be of value to aeronautical laboratorys held at NPS as well as future research and improvements to the Low Speed Wind Tunnel Facility.

VIII. RECOMMENDATIONS

This thesis project was designed and built as a steady state data acquisition system. The next logical step for hotwire anemometry research is to take advantage of its fast response time. Hardware and software additions should be made so the IBM PC-AT computer can handle data inputs of up to 100 kilohertz. With these changes installed, a real time analysis of wake turbulence, velocity vector components, and turbulence intensity can be realized. One short term improvement that would greatly enhance current measurement capabilities is to replace the Hewlett Packard Digital Multi-Meter with a voltmeter capable of outputting a mean voltage reading. With this change incorporated, all velocity profiles up to the stall region would be smooth and continuous.

Hewlett Packard now produces a software package that allows their data acquisition equipment to be compatible with Quick Basic (a compiled form of Basic). Quick Basic is approximately ten times faster in execution than the Advanced Basic (BASICA) used in this thesis project. If CALIB and PART2 were executed in Quick Basic then hundreds of measurements per data point would become a reality since time constraints would become negligible. This would eliminate the need for a voltmeter with mean value output capability.

The traverse assembly is one of the greatest limitations to experimental efficiency. The current configuration requires two people to coordinate traverse movement through the use of hand signals (to compensate for tunnel noise) or one person to repeatedly move back and forth between the wind tunnel test section and the computer. A motorized traverse assembly that can be computer controlled with no loss in accuracy is highly recommended.

Another system deficiency resides in the turntable positioning unit for angle of incidence control. The chain drive assembly has play in its movement that cannot be compensated for in a computer program. The relay/DC motor system is imprecise and must rely on software timing for correct positioning. The signal amplifier will waiver as much as 0.2 degrees after turntable motion has ceased. To compensate for this lack of precision it is recommended that angles of incidence be entered into PART2 one at a time if multiple experiments are to be performed. As long as the tunnel or hotwire system is not shut down, PART2 can be restarted repeatedly without altering system calibration.

The Hewlett Packard Laser Jet Printer also contributes to inefficiency in the laboratory. Graphical outputs each take between 4-6 minutes for hardcopy output. During this time the computer is continually sending data to the printer and is not available for other use. A faster printer or a printer buffer that would free the computer for other uses would eliminate unproductive waiting during classroom demonstrations. This upgrade is desireable for laboratory work as well and is highly recommended.

System acquisition for this thesis project seems to have been performed in a sporadic manner. Large amounts of money were invested in top quality equipment (ie, the HP Data Acquisition System and the Para-Scientific Pressure Computer) but their performance becomes marginal when used as a cohesive data acquisition unit. For example, the Para-Scientific Pressure Computer provides a 50 pin digital output to supply pressure data with a +/- 0.00001 PSI accuracy. The HP Digital Input Port can only accept 16 data pins, therefore the precision of the Pressure Computer is now reduced to +/- 0.002 PSI. This accuracy is only a marginal improvement over the analog pressure transducers previously installed in the wind tunnel test section. It is highly recommended that any new equipment purchased for upgrading present wind tunnel capabilities be carefully scrutinized for compatibility with equipment already in use.

APPENDIX A

Sample Calculations of Test Section Correction Factors

Solid Blocking, Esb

$$Esb = \frac{K1 * \text{Model Volume}}{C^{1.5}}$$

where K1 = 0.52 for a model spanning the test section vertically

C = tunnel test section area (8.75 ft^2)

Model Volume = .7 * max thickness * chord * span

Model Volume = 451.584 in^3

Model Volume = .2613 ft^3

$$Esb = \frac{.52 * .2613}{8.75^{1.5}}$$

Esb = 0.00525

Wake Blocking, Ewb

$$Ewb = (c/2W) * Cdu$$

where c = model chord = 12 in

W = test section width = 45 in

Cdu = uncorrected drag coefficient determined for each experimental profile

Corrected Drag Coefficient, Cd

$$Cd = Cdu * (1- 3Esb - 2Ewb)$$

APPENDIX B

Program Listing: Shell Program and CALIB

```
1 DEF SEG:CLEAR ,&HFE00: KEY OFF: COLOR 1,7,7: LOCATE 12,26:
PRINT "INITIALIZING HP INSTRUMENTS": GOTO 4
2 GOTO 1000 ' User program
3 GOTO 900 ' Error handling
4 I=&HFE00 ' Copyright Hewlett-Packard 1984,1985
5 PCIB.DIR$=ENVIRON$("PCIB"): CLS: COLOR 1,7,7: LOCATE
12,26: PRINT "INITIALIZING HP INSTRUMENTS"
6 I$=PCIB.DIR$+"\PCIBILC.BLD"
7 BLOAD I$,I
8 CALL I(PCIB.DIR$,I%,J%):PCIB.SEG=I%
9 IF J%=0 THEN GOTO 13
10 PRINT "Unable to load.";
11 PRINT "   (Error #";J%;")"
12 END
13 '
14 DEF SEG=PCIB.SEG:O.S=5:C.S=10:I.V=15
15 I.C=20:L.P=25:LD.FILE=30
16 GET.MEM=35:L.S=40:PANELS=45:DEF.ERR=50
17 PCIB.ERR$=STRING$(64,32) : PCIB.NAME$=STRING$(16,32)
18 CALL DEF.ERR(PCIB.ERR,PCIB.ERR$,PCIB.NAME$,PCIB.GLBERR) :
PCIB.BASERR=255
19 ON ERROR GOTO 3
20 J=-1
21 I$=PCIB.DIR$+"\PCIB.SYN"
22 CALL O.S(I$)
23 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
24 I=0
25 CALL
I.V(I,READ.REGISTER,READ.SELFID,DEFINE,INITIALIZE.SYSTEM)
26 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
27 CALL
I.V(I,ENABLE.SYSTEM,DISABLE.SYSTEM,INITIALIZE,POWER.ON)
28 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
29 CALL I.V(I,MEASURE,OUTPUT,START,HALT)
30 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
31 CALL
I.V(I,ENABLE.INT.TRIGGER,DISABLE.INT.TRIGGER,ENABLE.OUTPUT,D
ISABLE.OUTPUT)
32 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
33 CALL I.V(I,CHECK.DONE,GET.STATUS,SET.FUNCTION,SET.RANGE)
34 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
35 CALL I.V(I,SET.MODE,WRITE.CAL,READ.CAL,STORE.CAL)
36 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
37 CALL I.V(I,DELAY,SAVE.SYSTEM,J,J)
38 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
```

```
39 I=1
40 CALL I.V(I,SET.GATETIME,SET.SAMPLES,SET.SLOPE,SET.SOURCE)
41 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
42 CALL I.C(I,FREQUENCY,AUTO.FREQ,PERIOD,AUTO.PER)
43 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
44 CALL I.C(I,INTERVAL,RATIO,TOTALIZE,R100MILLI)
45 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
46 CALL I.C(I,R1,R10,R100,R1KILO)
47 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
48 CALL I.C(I,R10MEGA,R100MEGA,CHAN.A,CHAN.B)
49 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
50 CALL I.C(I,POSITIVE,NEGATIVE,COMN,SEPARATE)
51 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
52 I=2
53 I=3
54 CALL I.V(I,ZERO.OHMS,SET.SPEED,J,J)
55 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
56 CALL I.C(I,DCVOLTS,ACVOLTS,OHMS,R200MILLI)
57 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
58 CALL I.C(I,R2,R20,R200,R2KILO)
59 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
60 CALL I.C(I,R20KILO,R200KILO,R2MEGA,R20MEGA)
61 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
62 CALL I.C(I,AUTOM,R2.5,R12.5,J)
63 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
64 I=4
65 CALL
I.V(I,SET.COMPLEMENT,SET.DRIVER,OUTPUT.NO.WAIT,ENABLE.HANDSH
AKE)
66 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
67 CALL
I.V(I,DISABLE.HANDSHAKE,SET.THRESHOLD,SET.START.BIT,SET.NUM.
BITS)
68 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
69 CALL I.V(I,SET.LOGIC.SENSE,J,J,J)
70 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
71 CALL I.C(I,POSITIVE,NEGATIVE,TWOS,UNSIGNED)
72 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
73 CALL I.C(I,OC,TTL,R0,R1)
74 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
75 CALL I.C(I,R2,R3,R4,R5)
76 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
77 CALL I.C(I,R6,R7,R8,R9)
78 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
79 CALL I.C(I,R10,R11,R12,R13)
80 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
81 CALL I.C(I,R14,R15,R16,J)
82 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
83 I=6
84 CALL
I.V(I,SET.FREQUENCY,SET.AMPLITUDE,SET.OFFSET,SET.SYMMETRY)
```

```
85 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
86 CALL I.V(I,SET.BURST.COUNT,J,J,J)
87 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
88 CALL I.C(I,SINE,SQUARE,TRIANGLE,CONTINUOUS)
89 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
90 CALL I.C(I,GATED,BURST,J,J)
91 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
92 I=7
93 CALL
I.V(I,AUTOSCALE,CALIBRATE,SET.SENSITIVITY,SET.VERT.OFFSET)
94 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
95 CALL
I.V(I,SET.COUPLING,SET.POLARITY,SET.SWEEPSPEED,SET.DELAY)
96 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
97 CALL
I.V(I,SET.TRIG.SOURCE,SET.TRIG.SLOPE,SET.TRIG.LEVEL,SET.TRIG
.MODE)
98 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
99 CALL
I.V(I,GET.SINGLE.WF,GET.TWO.WF,GET.VERT.INFO,GET.TIMEBASE.IN
FO)
100 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
101 CALL
I.V(I,GET.TRIG.INFO,CALC.WFVOLT,CALC.WFTIME,CALC.WF.STATS)
102 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
103 CALL
I.V(I,CALC.RISETIME,CALC.FALLTIME,CALC.PERIOD,CALC.FREQUENCY
)
104 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
105 CALL
I.V(I,CALC.PLUSWIDTH,CALC.MINUSWIDTH,CALC.OVERSHOOT,CALC.PRE
SHOOT)
106 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
107 CALL
I.V(I,CALC.PK.TO.PK,SET.TIMEOUT,SCOPE.START,MEASURE.SINGLE.W
F)
108 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
109 CALL I.V(I,MEASURE.TWO.WF,J,J,J)
110 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
111 CALL I.C(I,R10NANO,R100NANO,R1MICRO,R10MICRO)
112 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
113 CALL I.C(I,R100MICRO,R1MILLI,R10MILLI,R100MILLI)
114 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
115 CALL I.C(I,R1,R10,R20NANO,R200NANO)
116 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
117 CALL I.C(I,R2MICRO,R20MICRO,R200MICRO,R2MILLI)
118 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
119 CALL I.C(I,R20MILLI,R200MILLI,R2,R20)
120 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
121 CALL I.C(I,R50NANO,R500NANO,R5MICRO,R50MICRO)
122 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
```

```
123 CALL I.C(I,R500MICRO,R5MILLI,R50MILLI,R500MILLI)
124 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
125 CALL I.C(I,R5,R50,CHAN.A,CHAN.B)
126 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
127 CALL I.C(I,EXTERNAL,POSITIVE,NEGATIVE,AC)
128 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
129 CALL I.C(I,DC,TRIGGERED,AUTO.TRIG,AUTO.LEVEL)
130 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
131 CALL I.C(I,X1,X10,STANDARD,AVERAGE)
132 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
133 I=8
134 CALL I.V(I,OPEN.CHANNEL,CLOSE.CHANNEL,J,J)
135 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
136 CALL C.S
137 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
138 I$=PCIB.DIR$+"\PCIB.PLD"
139 CALL L.P(I$)
140 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
141 I$="DMM.01":I=3:J=0:K=0:L=1
142 CALL DEFINE(DMM.01,I$,I,J,K,L)
143 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
144 I$="Func.Gen.01":I=6:J=0:K=1:L=1
145 CALL DEFINE(FUNC.GEN.01,I$,I,J,K,L)
146 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
147 I$="Scope.01":I=7:J=0:K=2:L=1
148 CALL DEFINE(SCOPE.01,I$,I,J,K,L)
149 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
150 I$="Counter.01":I=1:J=0:K=3:L=1
151 CALL DEFINE(COUNTER.01,I$,I,J,K,L)
152 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
153 I$="Dig.In.01":I=4:J=0:K=4:L=1
154 CALL DEFINE(DIG.IN.01,I$,I,J,K,L)
155 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
156 I$="Dig.Out.01":I=4:J=1:K=4:L=1
157 CALL DEFINE(DIG.OUT.01,I$,I,J,K,L)
158 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
159 I$="Relay.Act.01":I=8:J=0:K=5:L=1
160 CALL DEFINE(RELAY.ACT.01,I$,I,J,K,L)
161 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
162 I$="Relay.Mux.01":I=2:J=0:K=6:L=1
163 CALL DEFINE(RELAY.MUX.01,I$,I,J,K,L)
164 IF PCIB.ERR<>0 THEN ERROR PCIB.BASERR
800 I$=ENVIRON$("PANELS")+"\PANELS.EXE"
801 CALL L.S(I$)
899 GOTO 2
900 IF ERR=PCIB.BASERR THEN GOTO 903
901 PRINT "BASIC error #";ERR;" occurred in line ";ERL
902 STOP
903 TMPERR=PCIB.ERR:IF TMPERR=0 THEN TMPERR=PCIB.GLBERR
904 PRINT "PC Instrument error #";TMPERR;" detected at line
";ERL
```

```
905 PRINT "Error: ";PCIB.ERR$
906 IF LEFT$(PCIB.NAME$,1)<>CHR$(32) THEN PRINT "Instrument:
";PCIB.NAME$
907 STOP
908 COMMON PCIB.DIR$,PCIB.SEG
909 COMMON LD.FILE,GET.MEM,PANELS,DEF.ERR
910 COMMON
PCIB.BASERR,PCIB.ERR,PCIB.ERR$,PCIB.NAME$,PCIB.GLBERR
911 COMMON
READ.REGISTER,READ.SELFID,DEFINE,INITIALIZE.SYSTEM,ENABLE.SY
STEM,DISABLE.SYSTEM,INITIALIZE,POWER.ON,MEASURE,OUTPUT,START
,HALT,ENABLE.INT.TRIGGER,DISABLE.INT.TRIGGER,ENABLE.OUTPUT,D
ISABLE.OUTPUT,CHECK.DONE,GET.STATUS
912 COMMON
SET.FUNCTION,SET.RANGE,SET.MODE,WRITE.CAL,READ.CAL,STORE.CAL
,DELAY,SAVE.SYSTEM,SET.GATETIME,SET.SAMPLES,SET.SLOPE,SET.SO
URCE,ZERO.OHMS,SET.SPEED,SET.COMPLEMENT,SET.DRIVER,OUTPUT.NO
.WAIT,ENABLE.HANDSHAKE,DISABLE.HANDSHAKE
913 COMMON
SET.THRESHOLD,SET.START.BIT,SET.NUM.BITS,SET.LOGIC.SENSE,SET
.FREQUENCY,SET.AMPLITUDE,SET.OFFSET,SET.SYMMETRY,SET.BURST.C
OUNT,AUTOSCALE,CALIBRATE,SET.SENSITIVITY,SET.VERT.OFFSET,SET
.COUPLING,SET.POLARITY,SET.SWEEPSPEED
914 COMMON
SET.DELAY,SET.TRIG.SOURCE,SET.TRIG.SLOPE,SET.TRIG.LEVEL,SET.
TRIG.MODE,GET.SINGLE.WF,GET.TWO.WF,GET.VERT.INFO,GET.TIMEBAS
E.INFO,GET.TRIG.INFO,CALC.WFVOLT,CALC.WFTIME,CALC.WF.STATS,C
ALC.RISETIME,CALC.FALLTIME,CALC.PERIOD
915 COMMON
CALC.FREQUENCY,CALC.PLUSWIDTH,CALC.MINUSWIDTH,CALC.OVERSHOOT
,CALC.PRESHOOT,CALC.PK.TO.PK,SET.TIMEOUT,SCOPE.START,MEASURE
.SINGLE.WF,MEASURE.TWO.WF,OPEN.CHANNEL,CLOSE.CHANNEL
916 COMMON
FREQUENCY,AUTO.FREQ,PERIOD,AUTO.PER,INTERVAL,RATIO,TOTALIZE,
R100MILLI,R1,R10,R100,R1KILO,R10MEGA,R100MEGA,CHAN.A,CHAN.B,
POSITIVE,NEGATIVE,COMN,SEPARATE,DCVOLTS,ACVOLTS,OHMS,R200MIL
LI,R2,R20,R200,R2KILO,R20KILO,R200KILO
917 COMMON
R2MEGA,R20MEGA,AUTOM,R2.5,R12.5,POSITIVE,NEGATIVE,TWOS,UNSIG
NED,OC,TTL,R0,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14
,R15,R16,SINE,SQUARE,TRIANGLE,CONTINUOUS,GATED,BURST,R10NANO
,R100NANO,R1MICRO,R10MICRO,R100MICRO
918 COMMON
R1MILLI,R10MILLI,R100MILLI,R1,R10,R20NANO,R200NANO,R2MICRO,R
20MICRO,R200MICRO,R2MILLI,R20MILLI,R200MILLI,R2,R20,R50NANO,
R500NANO,R5MICRO,R50MICRO,R500MICRO,R5MILLI,R50MILLI,R500MIL
LI,R5,R50,CHAN.A,CHAN.B,EXTERNAL,POSITIVE
919 COMMON
NEGATIVE,AC,DC,TRIGGERED,AUTO.TRIG,AUTO.LEVEL,X1,X10,STANDAR
D,AVERAGE
```

```
920 COMMON
DMM.01,FUNC.GEN.01,SCOPE.01,COUNTER.01,DIG.IN.01,DIG.OUT.01,
RELAY.ACT.01,RELAY.MUX.01
999 'End PCIB Program Shell
1000 ' PROGRAM "CALIB"RATE
1010 OPTION BASE 1
1020 WANTAOA = 0
1030 '
1040 '
1050 ' SET HP BOXES TO INITIAL SETTINGS FOR DATA ACQUISITION
1060 '
1070 '
1080 FILE$ = "HOTWIRE.HPC"
1090 CALL INITIALIZE.SYSTEM(FILE$)
1100 CALL ENABLE.SYSTEM
1110 CALL SET.FUNCTION(DMM.01,DCVOLTS)
1120 CALL SET.RANGE(DMM.01,AUTOM)
1130 CALL SET.SPEED(DMM.01,R2.5)
1140 CALL ENABLE.INT.TRIGGER(DMM.01)
1141 CALL SET.NUM.BITS(DIG.IN.01,R16)
1142 LEVEL = 5
1143 CALL SET.THRESHOLD(DIG.IN.01,LEVEL)
1150 '
1160 '
1170 ' PRINT FIRST SCREEN
1180 '
1190 '
1200 BEEP
1210 CLS
1220 COLOR 4: LOCATE 5,14: PRINT
"****************************************************"
1230 LOCATE 9,14: PRINT "********************
********************"
1240 COLOR 1: LOCATE 9,37: PRINT "PART 1": LOCATE 10,33:
COLOR 4: PRINT "**************"
1250 LOCATE 6,14: PRINT "*": LOCATE 7,14: PRINT "*": LOCATE
8,14: PRINT "*"
1260 LOCATE 6,65: PRINT "*": LOCATE 7,65: PRINT "*": LOCATE
8,65: PRINT "*"
1270 COLOR 1: LOCATE 7,18: PRINT "WELCOME TO THE HOT WIRE
FLOW MEASUREMENT LAB"
1280 LOCATE 12,27: PRINT "THESIS BY LT GREG DOREMUS"
1290 LOCATE 15,20: PRINT "THIS PROGRAM IS USED IN
CALIBRATING THE "
1300 LOCATE 17,13: PRINT "HOT WIRE SYSTEM FOR WIND TUNNEL
AIR FLOW MEASUREMENTS"
1310 COLOR 4: LOCATE 21,24
1320 INPUT "ENTER 1 TO CONTINUE, 2 TO ABORT";ANS1
1330 IF ANS1 <> 2 GOTO 1340 ELSE 5950
1340 CLS: LOCATE 10,36: COLOR 20: PRINT "WAIT...": COLOR 1
```

```
1350 LOCATE 14,22: PRINT "AIRFOIL IS BEING POSITIONED TO
ZERO"
1360 LOCATE 16,20: PRINT "DEGREES AOA FOR CALIBRATION
MEASUREMENTS"
1370 GOSUB 4940
1380 GOSUB 5070
1390 '
1400 '
1410 ' PRINT SECOND SCREEN
1420 '
1430 '
1440 CLS: LOCATE 6,10: PRINT "TO START THIS EXPERIMENT YOU
MUST FIRST SET UP THE HOT WIRE"
1450 LOCATE 8,12: PRINT "SYSTEM. THE HOT WIRE LABORATORY
MANUAL WILL EXPLAIN THIS"
1460 LOCATE 10,6: PRINT "PROCEDURE IN DETAIL. ONCE THE
SYSTEM SET UP IS TO YOUR SATISFACTION"
1470 LOCATE 12,10: PRINT "YOU MUST MANUALLY START THE WIND
TUNNEL AND SET THE SLOWEST"
1480 LOCATE 14,16: PRINT "TUNNEL VELOCITY YOU WILL USE IN
YOUR EXPERIMENT."
1490 COLOR 4: LOCATE 20,24: PRINT "ENTER 1 TO CONTINUE, 2 TO
ABORT"
1500 LOCATE 21,28: INPUT "OR 3 TO GO BACK ONE PAGE";ANS2
1510 COLOR 1: ON ANS2 GOTO 1560,5950,1210
1520 '
1530 ' PRINT THIRD SCREEN - START WIND TUNNEL
1540 '
1550 '
1560 CLS: LOCATE 10,22: PRINT "IF HOTWIRE SYSTEM IS SET UP
PROPERLY"
1570 COLOR 4: LOCATE 12,29: PRINT "START WIND TUNNEL NOW"
1580 LOCATE 20,24: PRINT "ENTER 1 TO CONTINUE, 2 TO ABORT"
1590 LOCATE 21,28: INPUT "OR 3 TO GO BACK ONE PAGE"; ANS3
1600 COLOR 1: ON ANS3 GOTO 1610,5950,1440
1610 '
1620 '
1630 ' PRINT FOURTH SCREEN
1640 '
1650 '
1660 CLS: LOCATE 8,13: PRINT "A SERIES OF WIND VELOCITY
MEASUREMENTS IS TO BE TAKEN"
1670 LOCATE 10,12: PRINT "AT WIND SPEEDS VARYING FROM THE
SLOWEST TUNNEL VELOCITY"
1680 LOCATE 12,11: PRINT "TO BE USED IN YOUR EXPERIMENT UP
TO THE HIGHEST VELOCITY."
1690 LOCATE 14,15: PRINT "A MINIMUM OF 10 READINGS SHOULD BE
TAKEN TO CREATE"
1700 LOCATE 16,29: PRINT "THE CALIBRATION CURVE."
1710 COLOR 4: LOCATE 20,24: PRINT "ENTER 1 TO CONTINUE, 2 TO
ABORT"
```

```
1720 LOCATE 21,28: INPUT "OR 3 TO GO BACK ONE PAGE";ANS4
1730 COLOR 1: ON ANS4 GOTO 1790,5950,1560
1740 '
1750 '
1760 ' ENTER DATA POINTS FOR CALIBRATION CURVE
1770 '
1780 '
1790 CLS: LOCATE 12,12: PRINT "ENTER THE
YOU WILL TAKE TO CREATE"
1800 COLOR 4: LOCATE 12,22: PRINT "NUMBER OF DATA POINTS":
COLOR 1
1810 LOCATE 14,19: INPUT "THE CALIBRATION CURVE FOR THIS
EXPERIMENT.";POINTS
1820 CLS
1830 TEST = 1
1840 CALL OUTPUT(RELAY.MUX.01,TEST)
1850 DIM Z(POINTS,2), X(POINTS), Y(POINTS), C(10), D(10),
KNOTS(POINTS,2), DEC(16)
1860 '
1870 '
1880 ' OPEN DATA FILES
1890 '
1900 '
1910 TIME$ = "00:00:00"
1920 COLOR 4
1930 LOCATE 14,18: PRINT "CREATING DATA FILES FOR
CALIBRATION VOLTAGES"
1940 COLOR 1
1950 OPEN "HOTWIRE.DAT" FOR OUTPUT AS #1
1960 CLOSE #1
1970 TIME = VAL(RIGHT$(TIME$,2))
1980 IF TIME < 5 GOTO 1970 ELSE 1990
1990 '
2000 '
2010 ' START DATA ACQUISITION
2020 '
2030 '
2040 READING = 0: CHANNEL = 0
2050 FOR READING = 1 TO POINTS
2051   CLS: LOCATE 10,15: PRINT "DEPRESS BUTTON    ON THE
PARASCIENTIFIC TRANSDUCER"
2052   COLOR 4: LOCATE 10,30: PRINT "P1": COLOR 1
2053   LOCATE 12,15: PRINT "MAKE SURE NO OTHER FUNCTION
BUTTONS ARE DEPRESSED"
2054   COLOR 4: LOCATE 21,28: PRINT "PRESS ENTER TO
CONTINUE"
2055   A$ = INKEY$: IF A$ = "" THEN 2055
2060   CLS: LOCATE 8,17: COLOR 1
2070   IF READING = 1 GOTO 2080 ELSE 2100
2080   PRINT "IS THE PROBE POSITIONED FOR YOUR FIRST
READING?"
```

```
2090   GOTO 2110
2100   LOCATE 8,19: PRINT "IS THE PROBE POSITIONED FOR A NEW
READING?"
2110   COLOR 4: LOCATE 9,36: INPUT "1 = YES";ANS5
2120   IF ANS5 <> 1 GOTO 2060
2130   COLOR 1
2140   CLS: LOCATE 16,19: COLOR 1: PRINT "IS THE TUNNEL SET
FOR A NEW FLOW VELOCITY?"
2150 COLOR 4: LOCATE 17,36: INPUT "1 = YES";ANS6
2160   IF ANS6 <> 1 GOTO 2140
2170   CLS: COLOR 20: LOCATE 8,35: PRINT "STAND BY"
2180   COLOR 1: LOCATE 13,24: PRINT "DATA VALUES FOR HOT
WIRE VOLTAGE"
2190   LOCATE 15,25: PRINT "AND TEST SECTION VELOCITY ARE"
2200   LOCATE 17,30: PRINT "NOW BEING COLLECTED"
2210     FOR CHANNEL = 1 TO 2
2220       ON CHANNEL GOTO 2230,2342
2230         CALL OUTPUT(RELAY.MUX.01,CHANNEL)
2240         CALL SET.RANGE(DMM.01,R20)
2250           I = 0: SUM1 = 0: FOR I = 1 TO 10
2260             IF I = 1 GOTO 2270 ELSE 2290
2270               TIME = .5
2280               CALL DELAY(TIME)
2290             CALL MEASURE(DMM.01,C(I))
2300             SUM1 = SUM1 + ABS(C(I))
2310           NEXT I
2320         AVG1 = SUM1/10
2330         Z(READING,CHANNEL) = AVG1
2340       GOTO 2780
2342           I = 0
2344           SUM2 = 0
2370         FOR I = 1 TO 10
2410           CALL MEASURE(DIG.IN.01,D(I))
2420           SUM2 = SUM2 + D(I)
2430         NEXT I
2440         DN = SUM2/10
2442         GOSUB 5480
2450         Z(READING,CHANNEL) = PRESS
2460         GOSUB 4270
2470         GOSUB 4700
2480         Z(READING,CHANNEL) = VEL
2490         KNOTS(READING,CHANNEL) = KTS
2500         BEEP: CLS: LOCATE 2,32: PRINT "CALIBRATION
DATA"
2510         LOCATE 4,17: PRINT "HOT WIRE VOLTAGE"
2520         LOCATE 4,46: PRINT "WIND TUNNEL VELOCITY":
COLOR 4
2530         LOCATE 6,19: PRINT USING "#.####";Z(READING,1):
COLOR 1
2540         LOCATE 6,26: PRINT "VOLTS": COLOR 4
```

```
2550            LOCATE 6,41: PRINT USING
"###.###";Z(READING,2): COLOR 1
2560            LOCATE 6,49: PRINT "FT/SEC": COLOR 4
2570            LOCATE 6,58: PRINT USING
"###.###";KNOTS(READING,2): COLOR 1
2580            LOCATE 6,66: PRINT "KNOTS": LOCATE 6,56: PRINT
"="
2590            LOCATE 8,15: PRINT "THIS IS DATA POINT
";READING;" OF ";POINTS
2600            LOCATE 10,32: PRINT "WIND TUNNEL DATA"
2610            LOCATE 12,15: PRINT "TUNNEL TEMP ="
2620            COLOR 4: LOCATE 12,29: PRINT USING "##.##";TF:
COLOR 1
2630            LOCATE 12,35: PRINT "DEG F"
2640            LOCATE 14,15: PRINT "TUNNEL STATIC PRESSURE ="
2650            COLOR 4: LOCATE 14,40: PRINT USING
"####.##";PSTAT: COLOR 1
2660            LOCATE 14,48: PRINT "LB/FT^2"
2670            LOCATE 16,15: PRINT "TUNNEL AIR DENSITY ="
2680            COLOR 4: LOCATE 16,36: PRINT USING
"#.#####";DENSITY: COLOR 1
2690            LOCATE 16,44: PRINT "LB/FT^3"
2700            LOCATE 18,15: PRINT "DYNAMIC PRESSURE ="
2710            COLOR 4: LOCATE 18,34: PRINT USING "##.##";Q:
COLOR 1
2720            LOCATE 18,40: PRINT "LB/FT^2"
2730            IF READING = POINTS GOTO 2750
2740            COLOR 20: LOCATE 21,24: PRINT "SET NEW FLOW
VELOCITY IN TUNNEL"
2750            COLOR 4: LOCATE 23,27: PRINT "PRESS ANY KEY TO
CONTINUE"
2760            A$ = INKEY$: IF A$ = "" THEN 2760
2770            COLOR 1
2780          NEXT CHANNEL
2790        NEXT READING
2800 BNUM1 = Z(1,2)
2810 BNUM2 = Z(POINTS,2)
2820 BNUM3 = KNOTS(1,2)
2830 BNUM4 = KNOTS(POINTS,2)
2840 '
2850 '
2860 ' STORE DATA IN DATA FILE LABELED "HOTWIRE.DAT"
2870 '
2880 '
2890 CLS: LOCATE 8,15: PRINT "DATA VALUES ARE BEING STORED
IN DATA FILE LABELED"
2900 COLOR 4: LOCATE 10,34: PRINT "HOTWIRE.DAT": COLOR 1
2910 I = 0: C1 = 0: C2 = 0: D1 = 0: D2 = 0
2920 FOR I = 1 TO POINTS
2930   X(I) = SQR(Z(I,2))
2940   Y(I) = (Z(I,1))^2
```

```
2950   C1 = C1 + X(I)
2960   C2 = C2 + X(I)^2
2970   D1 = D1 + Y(I)
2980   D2 = D2 + X(I)*Y(I)
2990 NEXT I
3000 OPEN "HOTWIRE.DAT" FOR APPEND AS #1
3010 J = 0: FOR J = 1 TO POINTS
3020   PRINT #1,USING"###.####";Y(J), X(J)
3030 NEXT J
3040 CLOSE #1
3050 '
3060 '
3070 ' SOLVE FOR SLOPE AND Y-INTERCEPT OF REGRESSION LINE
3080 '
3090 '
3100 A = (POINTS*D2-C1*D1)/(POINTS*C2-C1^2)
3110 IF ABS(A) < 1E-08 THEN A=0
3120 B = (D1-A*C1)/POINTS
3130 IF ABS(B) < 1E-08 THEN B=0
3140 '
3150 '
3160 ' PRINT SUMMARY OF CALIBRATION DATA ON SCREEN
3170 '
3180 '
3190 LOCATE 16,12: PRINT "WOULD YOU LIKE TO SEE A SUMMARY OF
YOUR CALIBRATION DATA"
3200 LOCATE 18,33: PRINT "ON THE SCREEN?": COLOR 4
3210 LOCATE 20,27: INPUT "ENTER 1 FOR YES, 0 FOR NO";ANS7
3220 IF ANS7 <> 0 GOTO 3230 ELSE 3640
3230 COLOR 1: CLS: LOCATE 2,32: PRINT "CALIBRATION DATA"
3240 LOCATE 4,9: PRINT "HOT WIRE VOLTAGE (VOLTS)"
3250 LOCATE 4,41: PRINT "WIND TUNNEL VELOCITY (FT/SEC)":
COLOR 4
3260 J = 0
3270 FOR J = 1 TO POINTS
3280   LOCATE J+5,18: PRINT USING"#.####";Z(J,1)
3290   LOCATE J+5,52: PRINT USING"###.###";Z(J,2)
3300 NEXT J
3310 '
3320 '
3330 ' SEND CALIBRATION DATA TO HP LASER JET PRINTER
3340 '
3350 '
3360 COLOR 1: LOCATE 22,20: PRINT "WOULD YOU LIKE A HARD
COPY OF THIS DATA?"
3370 COLOR 4: LOCATE 23,27: INPUT "ENTER 1 FOR YES, 2 FOR
NO";ANS8
3380 IF ANS8 <> 2 GOTO 3390 ELSE 3640
3390 CLS: LOCATE 12,26: PRINT "DATA SENT TO LASER PRINTER":
COLOR 1
3400 TIME$ = "00:00:00"
```

```
3410 TIME = VAL(RIGHT$(TIME$,2))
3420 IF TIME < 5 GOTO 3410 ELSE 3430
3430 LPRINT: LPRINT
3440 LPRINT TAB(32): LPRINT "CALIBRATION DATA"
3450 LPRINT
3460 LPRINT TAB(9): LPRINT "HOTWIRE VOLTAGE (VOLTS)
WIND TUNNEL VELOCITY"
3470 LPRINT TAB(42): LPRINT "(FT/SEC)              (KTS)"
3480 LPRINT
3490 J = 0
3500   FOR J = 1 TO POINTS
3510   LPRINT TAB(18);
3520   LPRINT USING"#.####";Z(J,1);
3530   LPRINT TAB(43);
3540   LPRINT USING"###.###";Z(J,2);
3550   LPRINT TAB(61);
3560   LPRINT USING"###.###";KNOTS(J,2)
3570 NEXT J
3580 LPRINT CHR$(12)
3590 '
3600 '
3610 ' PRINT VIEW PLOT OF CALIBRATION CURVE ON THE SCREEN
3620 '
3630 '
3640 COLOR 1: CLS: LOCATE 12,13: PRINT "WOULD YOU LIKE TO
SEE A PLOT OF THE CALIBRATION CURVE?"
3650 COLOR 4: LOCATE 14,27: INPUT "ENTER 1 FOR YES, 2 FOR
NO";ANS9
3660 COLOR 1: IF ANS9 <> 2 GOTO 3670 ELSE 3730
3670 CLS: SHELL "COMMAND/C HOT1.BAT"
3680 '
3690 '
3700 ' SEND HARD COPY OF CALIBRATION CURVE TO HP LASER JET
PRINTER
3710 '
3720 '
3730 COLOR 1: CLS: LOCATE 12,19: PRINT "WOULD YOU LIKE A
HARD COPY OF THIS GRAPH?"
3740 LOCATE 14,27: COLOR 4: INPUT "ENTER 1 FOR YES, 2 FOR
NO";ANS10
3750 COLOR 1: IF ANS10 <> 2 GOTO 3760 ELSE 3840
3760 CLS: COLOR 1: LOCATE 10,21: PRINT "THIS PROGRAM WILL
AUTOMATICALLY LEAVE"
3770 LOCATE 12,22: PRINT "BASICA AND ENTER A GRAPHICS
PROGRAM."
3780 LOCATE 14,14: PRINT "THE PLOT TAKES APPROXIMATELY 4-6
MINUTES TO COMPLETE"
3790 COLOR 4: LOCATE 21,27: PRINT "PRESS ANY KEY TO
CONTINUE"
3800 A$ = INKEY$: IF A$ = "" THEN 3800
3810 COLOR 1
```

```
3820 SHELL "HOT2.BAT"
3830 BEEP
3840 OPEN "CHAIN.DAT" FOR OUTPUT AS #1
3850   PRINT #1,USING"####.###"; A, B, BNUM1, BNUM2, BNUM3,
BNUM4, TF
3860 CLOSE #1
3870 '
3880 '
3890 ' PRINT CONCLUDING SCREEN AND SUMMARY OF CALIBRATION
RANGE
3900 '
3910 '
3920 CLS: LOCATE 8,14: PRINT "THIS CONCLUDES THE CALIBRATION
CURVE PORTION OF YOUR"
3930 LOCATE 10,10: PRINT "EXPERIMENT. THE HOT WIRE SYSTEM IS
NOW ACCURATELY CALIBRATED"
3940 LOCATE 12,13: PRINT "WITHIN A WIND TUNNEL VELOCITY
RANGE OF"
3950 COLOR 4: LOCATE 12,52: PRINT USING"###.###";Z(1,2)
3960 LOCATE 12,61: PRINT "FT/SEC": COLOR 1
3970 LOCATE 14,17: PRINT "(": LOCATE 14,18: PRINT USING
"###.###";KNOTS(1,2)
3980 LOCATE 14,26: PRINT "KTS) TO": COLOR 4
3990 LOCATE 14,34: PRINT USING "###.###";Z(POINTS,2): LOCATE
14,42: PRINT "FT/SEC": COLOR 1: LOCATE 14,49: PRINT "("
4000 LOCATE 14,50: PRINT USING "###.###";KNOTS(POINTS,2)
4010 LOCATE 14,58: PRINT "KTS)"
4020 COLOR 4: LOCATE 20,24: PRINT "ENTER 1 TO CONTINUE, 2 TO
ABORT"
4030 LOCATE 21,28: INPUT "OR 3 TO GO BACK ONE PAGE";ANS11
4040 ON ANS11 GOTO 4100,5950,3730
4050 '
4060 '
4070 ' PRINT LAST SCREEN AND CALIBRATION EQUATION
4080 '
4090 '
4100 CLS: LOCATE 8,30: PRINT "Y = "
4110 LOCATE 8,34: PRINT USING "##.###";A: LOCATE 8,40: PRINT
"X"
4120 IF B>=0 THEN LOCATE 8,42: PRINT "+": LOCATE 8,44: PRINT
USING "##.###";B ELSE LOCATE 8,42: PRINT "-": LOCATE 8,44:
PRINT USING "###.###";ABS(B)
4130 COLOR 1: LOCATE 10,10: PRINT "THIS IS THE STRAIGHT LINE
EQUATION OF THE CALIBRATION CURVE"
4140 LOCATE 12,19: PRINT "IT WILL AUTOMATICALLY BE ENTERED
IN PART 2"
4150 LOCATE 14,26: PRINT "OF YOUR HOT WIRE EXPERIMENT"
4160 COLOR 4: LOCATE 20,27: PRINT " ENTER 1 TO START PART
TWO"
4170 LOCATE 21,32: PRINT "ENTER 2 TO ABORT"
```

```
4180 LOCATE 22,25: INPUT "OR ENTER 3 TO GO BACK ONE
PAGE";ANS12
4190 COLOR 1
4200 ON ANS12 GOTO 4210,5950,3920
4210 LOAD "PART2",R
4220 '
4230 '
4240 ' SUBROUTINE FOR MEASURING WIND TUNNEL TEMPERATURE
4250 '
4260 '
4270 A1 = 38.709457#
4280 A2 = .037085566#
4290 A3 = 5.649552E-05
4300 B0 = .10086091#
4310 B1 = 25727.94369#
4320 B2 = -767345.8295#
4330 B3 = 78025595.81#
4340 B4 = -9247486589#
4350 B5 = 6.97688E+11
4360 B6 = -2.66192E+13
4370 B7 = 3.94078E+14
4380 B8 = 0
4390 B9 = 0
4400 REF.OUT = 8
4410 CALL OUTPUT(RELAY.MUX.01,REF.OUT)
4420 CALL ENABLE.OUTPUT(RELAY.MUX.01)
4430 CALL SET.FUNCTION(DMM.01,DCVOLTS)
4440 CALL SET.RANGE(DMM.01,R2)
4450 CALL DISABLE.INT.TRIGGER(DMM.01)
4460   COUNT = .5
4470   TIME$ = "00:00:00"
4480   IF TIMER < COUNT GOTO 4480 ELSE 4490
4490 CALL MEASURE(DMM.01,V)
4500 TR = 100*V
4510 ER = (TR*(A1+TR*(A2+TR*A3)))*10^-6
4520 CALL SET.RANGE(DMM.01,R200MILLI)
4530 TC = 7
4540 CALL OUTPUT(RELAY.MUX.01,TC)
4550   COUNT = .5
4560   TIME$ = "00:00:00"
4570   IF TIMER < COUNT GOTO 4570 ELSE 4580
4580 CALL MEASURE(DMM.01,ET)
4590 E = ER + ET
4600 Z = B5+E*(B6+E*(B7+E*(B8+E*B9)))
4610 T = B0+E*(B1+E*(B2+E*(B3+E*(B4+E*Z))))
4620 TF = (1.8*T) + 32
4630 RETURN
4640 '
4650 '
4660 ' SUBROUTINE FOR WIND TUNNEL DATA:
4670 '                  STATIC PRESS IN TEST SECTION
```

```
4680 '             AIR DENSITY IN TEST SECTION
4690 '             DYNAMIC PRESSURE IN TEST SECTION
4700 '             TEST SECTION VELOCITY, FT/SEC AND KTS
4710 '
4720 '
4722 CLS: COLOR 1: LOCATE 10,15: PRINT "DEPRESS BUTTON    ON
THE PARASCIENTIFIC TRANSDUCER"
4724 COLOR 4: LOCATE 10,30: PRINT "P2": COLOR 1
4726 LOCATE 12,15: PRINT "MAKE SURE NO OTHER FUNCTION
BUTTONS ARE DEPRESSED"
4728 COLOR 4: LOCATE 21,28: PRINT "PRESS ENTER TO CONTINUE"
4730 A$ = INKEY$: IF A$ = "" THEN 4730
4800 CALL MEASURE(DIG.IN.01,DN)
4802 GOSUB 5480
4810 PSTAT = PRESS*144
4820 TTEMP = TF + 459.67
4830 GASCONST = 53.3
4840 DENSITY = PSTAT/(GASCONST*TTEMP)
4850 DELTAP = Z(READING,CHANNEL)*144
4860 Q = DELTAP/.93
4870 IF Q < 1 THEN Q = 0
4880 GC = 32.174
4890 VEL = SQR(ABS((2*Q*GC)/DENSITY))
4900 KTS = VEL * .5924
4910 RETURN
4920 '
4930 '
4940 ' SUBROUTINE FOR READING AOA
4950 '
4960 '
4970 INPUT.CH = 5
4980 CALL OUTPUT(RELAY.MUX.01,INPUT.CH)
4990 CALL ENABLE.OUTPUT(RELAY.MUX.01)
5000 TIME = 3
5010 CALL DELAY(TIME)
5020 CALL MEASURE(DMM.01,AOA)
5030 HAVEAOA = (AOA-.003)*1000/10
5040 RETURN
5050 '
5060 '
5070 ' SUBROUTINE FOR AOA SELECTION
5080 '
5090 '
5100 WHILE (CINT(ABS(WANTAOA - HAVEAOA) * 10) / 10) >= .25
5110   ABSDIFF = ABS(WANTAOA - HAVEAOA)
5120   IF ABSDIFF > 36 THEN ABSDIFF = 36
5130   SWITCH.1 = 1
5140   SWITCH.2 = 2
5150   CALL OPEN.CHANNEL(RELAY.ACT.01,SWITCH.1)
5160   CALL OPEN.CHANNEL(RELAY.ACT.01,SWITCH.2)
5170   IF WANTAOA > HAVEAOA GOTO 5210 ELSE 5340
```

```
5180         '
5190         ' THIS PART OF THE SUBROUTINE MOVES THE TURNTABLE
FORWARD
5200         '
5210         IF (CINT(ABSDIFF*10)/10) >= .4 THEN TIMING = 100
ELSE TIMING = 1
5220         COUNT = ((ABSDIFF * TIMING) + 190) / 1000
5230         IF COUNT > 3 THEN COUNT = 3
5240         CALL CLOSE.CHANNEL(RELAY.ACT.01,SWITCH.1)
5250         CALL DELAY(COUNT)
5260         CALL OPEN.CHANNEL(RELAY.ACT.01,SWITCH.1)
5270           TIME = .5
5280           CALL DELAY(TIME)
5290         GOSUB 4940
5300         GOTO 5440
5310           '
5320           ' THIS PART OF THE SUBROUTINE MOVES THE TURNTABLE
IN REVERSE
5330           '
5340           IF (CINT(ABSDIFF*10)/10) >= .4 THEN TIMING = 100
ELSE TIMING = 1
5350           COUNT = ((ABSDIFF* TIMING) + 190)/1000
5360           IF COUNT > 3 THEN COUNT = 3
5370           CALL CLOSE.CHANNEL(RELAY.ACT.01,SWITCH.2)
5380           CALL DELAY(COUNT)
5390           CALL OPEN.CHANNEL(RELAY.ACT.01,SWITCH.2)
5400           TIME = .5
5410           CALL DELAY(TIME)
5420           GOSUB 4940
5430           GOTO 5440
5440 WEND
5450 RETURN
5460 '
5470 '
5480 ' SUBROUTINE FOR CONVERTING DECIMAL PRESSURE VALUES TO
NUMERIC VALUES
5490 '
5500 '
5510 COUNT = 0: PRESS = 0
5520 '
5530 ' DN IS THE MEASURED DECIMAL NUMBER
5540 ' DEC IS THE DECIMAL TRANSFORM MATRIX
5550 ' PRESS IS THE CONVERTED NUMERIC PRESSURE VALUE
5560 '
5570 DEC(16) = 10: DEC(15) = 8: DEC(14) = 4: DEC(13) = 2:
DEC(12) = 1
5580 DEC(11) = .8: DEC(10) = .4: DEC(9) = .2: DEC(8) = .1:
DEC(7) = .08
5590 DEC(6) = .04: DEC(5) = .02: DEC(4) = .01: DEC(3) =
8.000001E-03
5600 DEC(2) = .004: DEC(1) = .002
```

```
5610 '
5620 IF DN < 0 THEN PRESS = DEC(16) ELSE GOTO 5640
5630 COUNT = -32768!
5640 IF (DN-COUNT)/16384 >= 1 THEN PRESS = PRESS + DEC(15)
ELSE GOTO 5660
5650 COUNT = COUNT + 16384
5660 IF (DN-COUNT)/8192 >= 1 THEN PRESS = PRESS + DEC(14)
ELSE GOTO 5680
5670 COUNT = COUNT + 8192
5680 IF (DN-COUNT)/4096 >= 1 THEN PRESS = PRESS + DEC(13)
ELSE GOTO 5700
5690 COUNT = COUNT + 4096
5700 IF (DN-COUNT)/2048 >= 1 THEN PRESS = PRESS + DEC(12)
ELSE GOTO 5720
5710 COUNT = COUNT + 2048
5720 IF (DN-COUNT)/1024 >= 1 THEN PRESS = PRESS + DEC(11)
ELSE GOTO 5740
5730 COUNT = COUNT + 1024
5740 IF (DN-COUNT)/512 >= 1 THEN PRESS = PRESS + DEC(10)
ELSE GOTO 5760
5750 COUNT = COUNT + 512
5760 IF (DN-COUNT)/256 >= 1 THEN PRESS = PRESS + DEC(9) ELSE
GOTO 5780
5770 COUNT = COUNT + 256
5780 IF (DN-COUNT)/128 >= 1 THEN PRESS = PRESS + DEC(8) ELSE
GOTO 5800
5790 COUNT = COUNT + 128
5800 IF (DN-COUNT)/64 >= 1 THEN PRESS = PRESS + DEC(7) ELSE
GOTO 5820
5810 COUNT = COUNT + 64
5820 IF (DN-COUNT)/32 >= 1 THEN PRESS = PRESS + DEC(6) ELSE
GOTO 5840
5830 COUNT = COUNT + 32
5840 IF (DN-COUNT)/16 >= 1 THEN PRESS = PRESS + DEC(5) ELSE
GOTO 5860
5850 COUNT = COUNT + 16
5860 IF (DN-COUNT)/8 >= 1 THEN PRESS = PRESS + DEC(4) ELSE
GOTO 5880
5870 COUNT = COUNT + 8
5880 IF (DN-COUNT)/4 >= 1 THEN PRESS = PRESS + DEC(3) ELSE
GOTO 5900
5890 COUNT = COUNT + 4
5900 IF (DN-COUNT)/2 >= 1 THEN PRESS = PRESS + DEC(2) ELSE
GOTO 5920
5910 COUNT = COUNT + 2
5920 IF (DN-COUNT) = 1 THEN PRESS = PRESS + DEC(1)
5930 RETURN
5940 STOP
5950 CLS: SYSTEM
5960 END
```

APPENDIX C

Program Listing: PART2

```
1000 ' PROGRAM "PART2"
1010 CALL DEF.ERR(PCIB.ERR,PCIB.ERR$,PCIB.NAME$,PCIB.GLBERR)
1020 FILE$ = "HOTWIRE.HPC"
1030 CALL INITIALIZE.SYSTEM(FILE$)
1040 CALL ENABLE.SYSTEM
1050 CALL SET.FUNCTION(DMM.01,DCVOLTS)
1060 CALL SET.RANGE(DMM.01,AUTOM)
1070 CALL SET.SPEED(DMM.01,R2.5)
1080 CALL DISABLE.INT.TRIGGER(DMM.01)
1090 CALL ENABLE.OUTPUT(RELAY.MUX.01)
1100 CALL SET.NUM.BITS(DIG.IN.01,R16)
1110 LEVEL = 5
1120 CALL SET.THRESHOLD(DIG.IN.01,LEVEL)
1130 DIM WANTAOA(50), G(40), HWVOLT(100), POSITION(100),
V2(100), X(100),Y(100), PS.VEL(100), S.PR(100), Y2(100),
SPRDATA(100), DEC(16), M(100), Z(100)
1140 DIM YINT(100), Y2INT(100), F(70)
1150 CLS
1160 OPEN "CHAIN.DAT" FOR INPUT AS #1
1170   INPUT #1, A, B, BNUM1, BNUM2, BNUM3, BNUM4, TF
1180 CLOSE #1
1190 TCAL = TF
1200 ' TS IS THE TEMPERATURE OF THE HOTWIRE SENSOR (DEGREES
F)
1210 TS = 482
1220 '
1230 '
1240 ' PRINT INTRO SCREEN
1250 '
1260 '
1270 BEEP
1280 COLOR 4: LOCATE 5,14: PRINT
"****************************************************"
1290 LOCATE 9,14: PRINT "********************
********************"
1300 COLOR 1: LOCATE 9,37: PRINT "PART 2": LOCATE 10,33:
COLOR 4: PRINT "**************"
1310 LOCATE 6,14: PRINT "*": LOCATE 7,14: PRINT "*": LOCATE
8,14: PRINT "*"
1320 LOCATE 6,65: PRINT "*": LOCATE 7,65: PRINT "*": LOCATE
8,65: PRINT "*"
1330 COLOR 1: LOCATE 7,18: PRINT "WELCOME TO THE HOT WIRE
FLOW MEASUREMENT LAB"
1340 LOCATE 12,27: PRINT "THESIS BY LT GREG DOREMUS"
```

```
1350 LOCATE 15,20: PRINT "THIS PROGRAM IS USED TO ANALYZE
THE WAKE"
1360 LOCATE 16,18: PRINT "VELOCITY PROFILES OF THE NACA
66(215) - 216"
1370 LOCATE 17,24: PRINT "AIRFOIL USING HOTWIRE ANEMOMETRY"
1380 COLOR 4: LOCATE 21,24: INPUT "ENTER 1 TO CONTINUE, 2 TO
ABORT";AN1
1390 COLOR 1
1400 IF AN1 <> 2 GOTO 1440 ELSE 5180
1410 '
1420 '
1430 ' INITIAL AOA SELECTION: PREVIOUSLY STORED AOA VALUES
COME FROM DATA
1440 '                        FILE LABELED "AOA.DAT"
1450 '
1460 '
1470 CLS: LOCATE 9,9: PRINT "DO YOU WISH TO USE THE PREVIOUS
ANGLE OF INCIDENCE SELECTIONS?"
1480 COLOR 4
1490 LOCATE 20,27: INPUT "ENTER 1 FOR YES, 0 FOR NO";AN3
1500 COLOR 1
1510 IF AN3 <> 0 GOTO 1520 ELSE 1740
1520 LOCATE 13,28: PRINT "PREVIOUS AOA SELECTIONS:"
1530 COLOR 4
1540 OPEN "AOA.DAT" FOR INPUT AS #1
1550 SNUM = 1
1560 LOCATE 15,2
1570 IF EOF(1) THEN SNUM = -99
1580 IF SNUM = -99 GOTO 1630 ELSE 1590
1590   INPUT #1, WANTAOA(SNUM)
1600   IF SNUM = 1 THEN PRINT WANTAOA(SNUM); ELSE PRINT ",
";WANTAOA(SNUM);
1610   SNUM = SNUM + 1
1620 GOTO 1570
1630 CLOSE #1
1640 LOCATE 20,5: PRINT "
"
1650 LOCATE 21,8: PRINT " ENTER 1 TO USE THESE AOA
SELECTIONS, 2 TO ENTER NEW SELECTIONS"
1660 LOCATE 22,33: INPUT "OR 3 TO ABORT";AN4
1670 COLOR 1
1680 ON AN4 GOTO 2150, 1740, 5180
1690 '
1700 '
1710 ' AOA SELECTION - IF PREVIOUS AOA VALUES ARE NOT
DESIRED
1720 '
1730 '
1740 CLS: LOCATE 3,26: COLOR 4: PRINT "ANGLE OF INCIDENCE
SELECTION": COLOR 1
```

```
1750 LOCATE 8,22: PRINT "ENTER THE DESIRED ANGLE OF
INCIDENCE"
1760 LOCATE 10,25: PRINT "LIMITS ARE +18 TO -18 DEGREES
1770 LOCATE 12,19: PRINT "ENTER AS MANY AOA'S AS YOU WISH TO
MEASURE"
1780 LOCATE 15,31: PRINT "ENTER -99 TO STOP"
1790 LOCATE 20,22: PRINT "TOTAL NUMBER OF SELECTIONS
ENTERED:"
1800 COLOR 4
1810 SNUM = 0
1820 '
1830 ' THE VARIABLE SNUM STANDS FOR "SELECT NUMBER"
1840 '
1850 WHILE SELECT <> -99
1860     LOCATE 20,58: PRINT SNUM
1870     LOCATE 16,43: PRINT "      "
1880     LOCATE 16,37: INPUT "AOA:";SELECT
1890     SNUM = SNUM + 1
1900     WANTAOA(SNUM) = SELECT
1910 WEND
1920 LOCATE 22,11: INPUT "ENTER 1 TO STORE THESE FOR LATER
USE, 0 NOT TO STORE THEM";AN6
1930 IF AN6 <> 0 GOTO 1990 ELSE 2150
1940 '
1950 '
1960 ' STORE SELECTED AOA VALUES FOR FUTURE USE
1970 '
1980 '
1990 OPEN "AOA.DAT" FOR OUTPUT AS #1
2000 SNUM = 1
2010 WHILE WANTAOA(SNUM) <> -99
2020   PRINT #1, USING "###";WANTAOA(SNUM)
2030   SNUM = SNUM + 1
2040 WEND
2050 CLOSE #1
2060 LOCATE 22,11: PRINT "
"
2070 LOCATE 22,24: PRINT "ENTER 1 TO CONTINUE, 2 TO ABORT"
2080 LOCATE 23,28: INPUT "OR 3 TO GO BACK ONE PAGE"; AN7
2090 ON AN7 GOTO 2150, 5180, 1470
2100 '
2110 '
2120 ' POSITION TURNTABLE TO SELECTED AOA
2130 '
2140 '
2150 EXPER = 1
2160 WANTAOA = WANTAOA(EXPER)
2170 CLS: COLOR 20: LOCATE 10,35: PRINT "WAIT. . ."
2180 COLOR 1
2190 IF EXPER = 1 THEN GOTO 2200 ELSE GOTO 2240
```

```
2200 LOCATE 14,22: PRINT "AIRFOIL IS POSITIONING TO FIRST
AOA"
2210 GOSUB 6400
2220 GOSUB 6530
2230 GOTO 2270
2240 LOCATE 14,23: PRINT "AIRFOIL IS POSITIONING TO NEXT
AOA"
2250 GOSUB 6400
2260 GOSUB 6530
2270 '
2280 '
2290 ' ADJUST WIND TUNNEL VELOCITY WITHIN THE CALIBRATED
RANGE
2300 '
2310 '
2320 GOTO 2340
2330 COLOR 1: LOCATE 15,48: PRINT "UPDATING VELOCITY"
2340 GOSUB 6940
2350 GOSUB 7310
2360 COLOR 1
2370 CLS: LOCATE 6,15: PRINT "ADJUST TUNNEL SPEED WITHIN THE
CALIBRATED RANGE OF"
2380 COLOR 4: LOCATE 8,10: PRINT USING "###.###";BNUM1
2390 LOCATE 8,18: PRINT "FT/SEC": COLOR 1: LOCATE 8,25:
PRINT "("
2400 LOCATE 8,26: PRINT USING "###.###";BNUM3
2410 LOCATE 8,33: PRINT " KTS) TO": COLOR 4: LOCATE 8,42
2420 PRINT USING "###.###";BNUM2: LOCATE 8,50: PRINT
"FT/SEC"
2430 COLOR 1: LOCATE 8,57: PRINT "(": LOCATE 8,58: PRINT
USING "###.###";BNUM4
2440 LOCATE 8,66: PRINT"KTS)"
2450 LOCATE 12,28: PRINT "PRESENT TUNNEL VELOCITY:"
2460 COLOR 4
2470 LOCATE 15,41: PRINT "                          "
2480 LOCATE 14,33: PRINT USING "###.###";VEL
2490 LOCATE 14,41: PRINT "FT/SEC"
2500 LOCATE 16,33: PRINT USING "###.###";KTS
2510 LOCATE 16,41: PRINT "KNOTS"
2520 Q1 = Q
2530 LOCATE 20,23: PRINT "ENTER 1 TO UPDATE TUNNEL VELOCITY"
2540 COLOR 1: LOCATE 21,24: PRINT "(UPDATE TAKES APPROX 13
SECONDS)"
2550 COLOR 4: LOCATE 23,32: INPUT "OR 2 TO CONTINUE";AN10
2560 IF AN10 <> 2 GOTO 2330 ELSE 2590
2570 '
2580 '
2590 ' ADJUST POSITION OF THE TRAVERSE ASSEMBLY
2600 '
2610 '
2620 COLOR 1
```

```
2630 NUMBER = 0: READING = 0
2640 WHILE READING <> -99
2650    IF NUMBER = 0 GOTO 2660 ELSE 2690
2660    CLS: LOCATE 6,16: PRINT "MOVE THE PITOT-
STATIC/HOTWIRE TRAVERSE ASSEMBLY"
2670    LOCATE 8,16: PRINT "TO THE           POSITION FOR
YOUR FIRST READING"
2680    COLOR 4: LOCATE 8,23: PRINT "FOUR INCH": COLOR 1:
GOTO 2710
2690    CLS: LOCATE 6,16: PRINT "MOVE THE PITOT-
STATIC/HOTWIRE TRAVERSE ASSEMBLY"
2700    LOCATE 8,29: PRINT "FOR YOUR NEXT READING"
2710    LOCATE 12,27: PRINT "PRESENT POSITION:       INCH"
2720    INPUT.CH = 4
2730    CALL OUTPUT(RELAY.MUX.01,INPUT.CH)
2740    CALL ENABLE.OUTPUT(RELAY.MUX.01)
2750    CALL MEASURE(DMM.01,INCH)
2760    POSITION(NUMBER) = (INCH*1000)/1.053
2770    COLOR 4
2780    LOCATE 12,45: PRINT USING "##.##";POSITION(NUMBER)
2790    LOCATE 20,27: PRINT "PRESS ANY KEY WHEN YOU ARE"
2800    LOCATE 21,22: PRINT "THROUGH ADJUSTING TRAVERSE
POSITION"
2810    COLOR 1
2820    A$ = INKEY$: IF A$ = "" THEN GOTO 2750 ELSE 2830
2830    CLS: COLOR 20: LOCATE 10,36: PRINT "WAIT...": COLOR
1
2840    LOCATE 14,19: PRINT "HOT WIRE VOLTAGE DATA NOW BEING
COLLECTED"
2850    '
2860    '
2870    ' COLLECT HOTWIRE VOLTAGE DATA
2880    '
2890    '
2900    INPUT.CH = 1
2910    CALL OUTPUT(RELAY.MUX.01,INPUT.CH)
2920    I = 0: SUM1 = 0
2930    FOR I = 1 TO 40
2940       IF I = 1 GOTO 2950 ELSE 2970
2950       TIME = .5
2960       CALL DELAY(TIME)
2970       CALL MEASURE(DMM.01,G(I))
2980       SUM1 = SUM1 + ABS(G(I))
2990    NEXT I
3000    AVREAD1 = SUM1/40
3010    HWVOLT(NUMBER) = AVREAD1
3020    CORRFAC = (TS - TCAL)/(TS - TF)
3030    '
3040    '
3050    ' COLLECT PITOT-STATIC SYSTEM DATA
3060    '
```

```
3070    '
3080    GOSUB 6940
3090    GOSUB 7310
3100    ' PS.VEL IS THE TUNNEL VELOCITY AS MEASURED BY THE
PITOT-STATIC TUBE
3110    ' S.PR IS THE STATIC PRESSURE TAKEN FROM THE STATIC
PRESSURE PORT
3120    PS.VEL(NUMBER) = VEL
3130    '
3140    '
3150    ' GET VELOCITY PROFILE FROM HOTWIRE DATA
3160    '
3170    '
3180    X(NUMBER) = POSITION(NUMBER)
3190    ECORR = (HWVOLT(NUMBER)^2) * CORRFAC
3200    V2(NUMBER) = ((ECORR - B) / A)^2
3210    V1 = V2(0)
3220    Y(NUMBER) = V2(NUMBER)/V1
3230    IF WANTAOA >= 10 GOTO 3240 ELSE 3260
3240    IF Y(NUMBER) < .97 GOTO 3270 ELSE IF Y(NUMBER) >
1.03 GOTO 3270 ELSE Y(NUMBER) = 1
3250    GOTO 3270
3260    IF Y(NUMBER) < .985 GOTO 3270 ELSE IF Y(NUMBER) >
1.015 GOTO 3270 ELSE Y(NUMBER) = 1
3270    YINT(NUMBER) = Y(NUMBER) * (1 - Y(NUMBER))
3280    IF ABS(YINT(NUMBER)) <= .01 THEN YINT(NUMBER) = 0
3290    '
3300    '
3310    ' GET VELOCITY PROFILE FROM PITOT-STATIC SYSTEM DATA
3320    '
3330    '
3340    VINF = PS.VEL(0)
3350    Y2(NUMBER) = PS.VEL(NUMBER)/VINF
3360    IF Y2(NUMBER) < .985 GOTO 3370 ELSE IF Y2(NUMBER) >
1.015 GOTO 3370 ELSE Y2(NUMBER) = 1
3370    Y2INT(NUMBER) = (PS.VEL(NUMBER)/VINF)-
((PS.VEL(NUMBER)/VINF)^2)
3380    IF ABS(Y2INT(NUMBER)) <= .01 THEN Y2INT(NUMBER) = 0
3390    '
3400    '
3410    CLS: BEEP: COLOR 4: LOCATE 11,28: PRINT "PRESS ENTER
TO CONTINUE"
3420    LOCATE 13,14: INPUT "OR ENTER -99 IF YOU HAVE
COMPLETED YOUR LAST READING";READING
3430    IF READING = -99 GOTO 3450 ELSE 3440
3440    NUMBER = NUMBER + 1
3450    COLOR 1
3460 WEND
3470 EXPER = EXPER + 1
3480 CLS: LOCATE 10,15: PRINT "DATA VALUES ARE BEING STORED
IN DATA FILES LABELED"
```

```
3490 COLOR 4: LOCATE 12,35: PRINT "PSTAT.DAT"
3500 LOCATE 14,35: PRINT "HWIRE.DAT"
3510 COLOR 1
3520 OPEN "PSTAT.DAT" FOR OUTPUT AS #1
3530 J = 0
3540 FOR J = 1 TO NUMBER
3550   PRINT #1,USING"###.###";X(J), Y2(J)
3560 NEXT J
3570 CLOSE #1
3580 OPEN "HWIRE.DAT" FOR OUTPUT AS #1
3590 J = 0
3600 FOR J = 1 TO NUMBER
3610   PRINT #1,USING"###.###";X(J), Y(J)
3620 NEXT J
3630 CLOSE #1
3640 COLOR 1: CLS: LOCATE 1,31: PRINT "EXPERIMENTAL DATA"
3650 LOCATE 2,22: PRINT "ANGLE OF INCIDENCE (DEGREES) ="
3660 LOCATE 2,53: PRINT USING "###.#";WANTAOA
3670 LOCATE 3,42: PRINT "VELOCITY PROFILES"
3680 LOCATE 4,31: PRINT "HOTWIRE"
3690 LOCATE 4,56: PRINT "PITOT-STATIC"
3700 LOCATE 5,7: PRINT "POSITION (IN)   V2 (FT/SEC)
V2/VINF     V2 (FT/SEC)     V2/VINF"
3710 J = 0: COLOR 4
3720 FOR J = 1 TO NUMBER
3730   IF J >= 30 GOTO 4080 ELSE 3740
3740   IF J = 29 GOTO 3990 ELSE 3750
3750   IF J >= 16 GOTO 3920 ELSE 3760
3760   IF J = 15 GOTO 3830 ELSE 3770
3770   LOCATE J+6,11: PRINT USING "##.##";X(J)
3780   LOCATE J+6,25: PRINT USING "###.###";V2(J)
3790   LOCATE J+6,40: PRINT USING "#.####";Y(J)
3800   LOCATE J+6,53: PRINT USING "###.###";PS.VEL(J)
3810   LOCATE J+6,68: PRINT USING "#.###";Y2(J)
3820   GOTO 4140
3830   COLOR 1: LOCATE 22,23: PRINT "PRESS ANY KEY TO
DISPLAY MORE DATA"
3840   A$ = INKEY$: IF A$ = "" THEN 3840
3850   COLOR 1: CLS: LOCATE 1,31: PRINT "EXPERIMENTAL DATA"
3860   LOCATE 2,22: PRINT "ANGLE OF INCIDENCE (DEGREES) ="
3870   LOCATE 2,53: PRINT USING "###.#";WANTAOA
3880   LOCATE 3,42: PRINT "VELOCITY PROFILES"
3890   LOCATE 4,31: PRINT "HOTWIRE"
3900   LOCATE 4,56: PRINT "PITOT-STATIC"
3910   LOCATE 5,7: PRINT "POSITION (IN)   V2 (FT/SEC)
V2/VINF     V2 (FT/SEC)     V2/VINF"
3920   COLOR 4
3930   LOCATE J-8,11: PRINT USING "##.##";X(J)
3940   LOCATE J-8,25: PRINT USING "###.###";V2(J)
3950   LOCATE J-8,40: PRINT USING "#.####";Y(J)
3960   LOCATE J-8,53: PRINT USING "###.###";PS.VEL(J)
```

```
3970   LOCATE J-8,68: PRINT USING "#.####";Y2(J)
3980   GOTO 4140
3990   COLOR 1: LOCATE 22,23: PRINT "PRESS ANY KEY TO
DISPLAY MORE DATA"
4000   A$ = INKEY$: IF A$ = "" THEN 4000
4010   COLOR 1: CLS: LOCATE 1,31: PRINT "EXPERIMENTAL DATA"
4020   LOCATE 2,22: PRINT "ANGLE OF INCIDENCE (DEGREES) ="
4030   LOCATE 2,53: PRINT USING "###.#";WANTAOA
4040   LOCATE 3,42: PRINT "VELOCITY PROFILES"
4050   LOCATE 4,31: PRINT "HOTWIRE"
4060   LOCATE 4,56: PRINT "PITOT-STATIC"
4070   LOCATE 5,7: PRINT "POSITION (IN)   V2 (FT/SEC)
V2/VINF     V2 (FT/SEC)     V2/VINF"
4080   COLOR 4
4090   LOCATE J-22,11: PRINT USING "##.##";X(J)
4100   LOCATE J-22,25: PRINT USING "###.###";V2(J)
4110   LOCATE J-22,40: PRINT USING "#.####";Y(J)
4120   LOCATE J-22,53: PRINT USING "###.###";PS.VEL(J)
4130   LOCATE J-22,68: PRINT USING "#.####";Y2(J)
4140 NEXT J
4150 '
4160 '
4170 ' SEND DATA SUMMARY TO LASER PRINTER
4180 '
4190 '
4200 COLOR 1: LOCATE 22,20: PRINT "WOULD YOU LIKE A HARD
COPY OF THIS DATA?"
4210 COLOR 4: LOCATE 23,27: INPUT "ENTER 1 FOR YES, 2 FOR
NO";AN11
4220 IF AN11 <> 2 GOTO 4230 ELSE 4570
4230 CLS: LOCATE 12,26: PRINT "DATA SENT TO LASER PRINTER"
4240 COLOR 1: TIME$ = "00:00:00"
4250 TIME = VAL(RIGHT$(TIME$,2))
4260 IF TIME < 5 GOTO 4250 ELSE 4270
4270 LPRINT
4280 LPRINT TAB(31): LPRINT "EXPERIMENTAL DATA"
4290 LPRINT
4300 LPRINT TAB(22): LPRINT "ANGLE OF INCIDENCE (DEGREES)
=";
4310 LPRINT TAB(53);
4320 LPRINT USING "###.#";HAVEAOA
4330 LPRINT
4340 LPRINT TAB(42): LPRINT "VELOCITY PROFILES"
4350 LPRINT TAB(31): LPRINT "HOTWIRE                  PITOT-
STATIC"
4360 LPRINT TAB(7): LPRINT "POSITION (IN)    V2 (FT/SEC)
V2/VINF     V2 (FT/SEC)     V2/VINF"
4370 LPRINT
4380 J = 0
4390 FOR J = 0 TO NUMBER
4400   LPRINT TAB(11);
```

```
4410   LPRINT USING"##.##";X(J);
4420   LPRINT TAB(25);
4430   LPRINT USING"###.###";V2(J);
4440   LPRINT TAB(40);
4450   LPRINT USING"#.###";Y(J);
4460   LPRINT TAB(53);
4470   LPRINT USING"###.###";PS.VEL(J);
4480   LPRINT TAB(68);
4490   LPRINT USING"#.###";Y2(J)
4500 NEXT J
4510 LPRINT CHR$(12)
4520 '
4530 '
4540 ' PRINT VIEW PLOT OF HOTWIRE VELOCITY PROFILE
4550 '
4560 '
4570 COLOR 1
4580 CLS: LOCATE 12,9: PRINT "WOULD YOU LIKE TO SEE A PLOT
OF THE HOTWIRE VELOCITY PROFILE?"
4590 COLOR 4: LOCATE 14,27: INPUT "ENTER 1 FOR YES, 2 FOR
NO";AN12
4600 COLOR 1: IF AN12 <> 2 GOTO 4610 ELSE 4640
4610 CLS: SHELL"PART21.BAT"
4620 '
4630 '
4640 ' SEND HARD COPY OF HOTWIRE VELOCITY PROFILE TO LASER
PRINTER
4650 '
4660 '
4670 CLS: LOCATE 12,19: PRINT "WOULD YOU LIKE A HARD COPY OF
THIS GRAPH?"
4680 LOCATE 14,27: COLOR 4: INPUT "ENTER 1 FOR YES, 2 FOR
NO";AN13
4690 COLOR 1: IF AN13 <> 2 THEN GOSUB 4730 ELSE GOTO 4860
4700 GOTO 4830
4710 '
4720 '
4730 ' SUBROUTINE FOR INSTRUCTIONS ON LASER PRINTER
4740 '
4750 '
4760 CLS: COLOR 1: LOCATE 10,21: PRINT "THIS PROGRAM WILL
AUTOMATICALLY LEAVE"
4770 LOCATE 12,22: PRINT "BASICA AND ENTER A GRAPHICS
PROGRAM."
4780 LOCATE 14,14: PRINT "THE PLOT TAKES APPROXIMATELY 4-6
MINUTES TO COMPLETE"
4790 COLOR 4: LOCATE 21,27: PRINT "PRESS ANY KEY TO
CONTINUE"
4800 A$ = INKEY$: IF A$ = "" THEN 4800
4810 COLOR 1
4820 RETURN
```

```
4830 CLS: SHELL "PART22.BAT"
4840 '
4850 '
4860 ' PRINT VIEW PLOT OF PITOT-STATIC VELOCITY PROFILE
4870 '
4880 '
4890 CLS: LOCATE 12,7: PRINT "WOULD YOU LIKE TO SEE A PLOT
OF THE PITOT-STATIC VELOCITY PROFILE?"
4900 COLOR 4: LOCATE 14,27: INPUT "ENTER 1 FOR YES, 2 FOR
NO";AN14
4910 COLOR 1: IF AN14 <> 2 GOTO 4920 ELSE 4950
4920 CLS: SHELL "PART23.BAT"
4930 '
4940 '
4950 ' SEND HARD COPY OF PITOT-STATIC VELOCITY PROFILE TO
LASER PRINTER
4960 '
4970 '
4980 CLS: LOCATE 12,19: PRINT "WOULD YOU LIKE A HARD COPY OF
THIS GRAPH?"
4990 LOCATE 14,27: COLOR 4: INPUT "ENTER 1 FOR YES, 2 FOR
NO";AN15
5000 COLOR 1: IF AN15 <> 2 THEN GOSUB 4730 ELSE GOTO 5040
5010 CLS: SHELL "PART24.BAT"
5020 '
5030 '
5040 ' PRINT CONCLUDING SCREEN
5050 '
5060 '
5070 CLS: COLOR 4: LOCATE 6,12: PRINT
"********************************************************"
5080 LOCATE 10,12: PRINT
"********************************************************"
5090 LOCATE 7,12: PRINT "*": LOCATE 8,12: PRINT "*"
5100 LOCATE 9,12: PRINT "*": LOCATE 7,67: PRINT "*"
5110 LOCATE 8,67: PRINT "*": LOCATE 9,67: PRINT "*"
5120 COLOR 1: LOCATE 8,16: PRINT "THIS CONCLUDES THE HOT
WIRE FLOW MEASUREMENT LAB"
5130 COLOR 4: LOCATE 20,18: PRINT "ENTER 1 TO REPEAT THIS
LAB WITH THE NEXT AOA"
5140 LOCATE 21,15: PRINT "ENTER 2 TO CONTINUE WITH DRAG
COEFFICIENT ANALYSIS"
5150 LOCATE 22,20: INPUT "ENTER 3 TO EXIT BASICA AND RETURN
TO DOS";AN18
5160 ON AN18 GOTO 2160, 5190, 5180
5170 GOTO 5190
5180  CLS: SYSTEM
5190 X1 = X(1): X2 = X(NUMBER-3)
5200 I = 0
5210 FOR I = 0 TO NUMBER
5220   Y(I) = YINT(I)
```

```
5230 NEXT I
5240 GOSUB 8380
5250 PROFILE1 = Z
5260 CHORD = 12
5270 '
5280 ' CDU1 IS THE UNCORRECTED DRAG COEFFICIENT FOR THE
HOTWIRE SYSTEM
5290 '
5300 CDU1 = ((2/CHORD)*PROFILE1)
5310 '
5320 ' SBF IS THE SOLID BLOCKING FACTOR
5330 '
5340 SBF = .00525
5350 '
5360 ' WBF IS THE WAKE BLOCKAGE FACTOR
5370 WBF = .13333 * CDU1
5380 '
5390 ' CD1 IS THE CORRECTED DRAG COEFFICIENT FOR THE HOTWIRE
DATA ANALYSIS
5400 '
5410 CD1 = CDU1*(1-(3*SBF)-(2*WBF))
5420 '
5430 ' FIND DRAG COEFFICIENT FOR THE PITOT-STATIC SYSTEM
5440 '
5500 I = 0
5510 FOR I = 0 TO NUMBER
5520   Y(I) = 0
5530 NEXT I
5540 I = 0
5550 FOR I = 0 TO NUMBER
5560   Y(I) = Y2INT(I)
5570 NEXT I
5580 IF Y(NUMBER) < .01 THEN Y(NUMBER) = 0
5590 GOSUB 8380
5600 PROFILE3 = Z
5610 '
5620 ' CDU2 IS THE UNCORRECTED DRAG COEFFICIENT FOR THE
PITOT-STATIC SYSTEM
5630 '
5640 CDU2 = ((2/CHORD)*PROFILE3)
5650 WBF = .13333*CDU2
5660 '
5670 ' CD2 IS THE CORRECTED DRAG COEFFICIENT FOR THE PITOT-
STATIC SYSTEM
5680 '
5690 CD2 = CDU2*(1-(3*SBF)-(2*WBF))
5700 '
5710 '
5720 ' PRINT DRAG COEFFICIENT SUMMARY ON SCREEN
5730 '
5740 '
```

```
5750 COLOR 1: CLS: LOCATE 2,28: PRINT "DRAG COEFFICIENT
SUMMARY"
5760 LOCATE 4,26: PRINT "AIRFOIL AT        DEGREES AOA"
5770 LOCATE 4,37: COLOR 4: PRINT USING "###.#";HAVEAOA
5780 COLOR 1: LOCATE 7,33: PRINT "HOTWIRE SYSTEM"
5790 LOCATE 9,28: PRINT "UNCORRECTED Cd ="
5800 LOCATE 11,29: PRINT "CORRECTED Cd ="
5810 COLOR 4: LOCATE 9,45: PRINT USING "#.####";CDU1
5820 LOCATE 11,44: PRINT USING "#.####";CD1
5830 COLOR 1: LOCATE 14,30: PRINT "PITOT-STATIC SYSTEM"
5840 LOCATE 16,28: PRINT "UNCORRECTED Cd ="
5850 LOCATE 18,29: PRINT "CORRECTED Cd ="
5860 COLOR 4: LOCATE 16,45: PRINT USING "#.####";CDU2
5870 LOCATE 18,44: PRINT USING "#.####";CD2
5880 LOCATE 21,17: PRINT "WOULD YOU LIKE A HARD COPY OF THE
Cd SUMMARY?"
5890 LOCATE 22,27: INPUT "ENTER 1 FOR YES, 2 FOR NO";AN19
5900 IF AN19 <> 2 GOTO 5930 ELSE 6290
5910 '
5920 '
5930 ' SEND DRAG SUMMARY TO LASER PRINTER
5940 '
5950 '
5960 LPRINT
5970 LPRINT TAB(28): LPRINT "DRAG COEFFICIENT SUMMARY"
5980 LPRINT
5990 LPRINT
6000 LPRINT TAB(22): LPRINT "ANGLE OF INCIDENCE (DEGREES)
=";
6010 LPRINT TAB(53);
6020 LPRINT USING "###.#";HAVEAOA
6030 LPRINT
6040 LPRINT
6050 LPRINT
6060 LPRINT TAB(33): LPRINT "HOTWIRE SYSTEM"
6070 LPRINT
6080 LPRINT TAB(28): LPRINT "UNCORRECTED Cd =";
6090 LPRINT TAB(45);
6100 LPRINT USING "#.####";CDU1
6110 LPRINT
6120 LPRINT TAB(29): LPRINT "CORRECTED Cd =";
6130 LPRINT TAB(44);
6140 LPRINT USING "#.####";CD1
6150 LPRINT
6160 LPRINT
6170 LPRINT TAB(30): LPRINT "PITOT-STATIC SYSTEM"
6180 LPRINT
6190 LPRINT TAB(28):LPRINT "UNCORRECTED Cd =";
6200 LPRINT TAB(45);
6210 LPRINT USING "#.####";CDU2
6220 LPRINT
```

```
6230 LPRINT TAB(29): LPRINT "CORRECTED Cd =";
6240 LPRINT TAB(44);
6250 LPRINT USING "#.####";CD2
6260 LPRINT CHR$(12)
6270 '
6280 '
6290 LOCATE 21,16: PRINT "
"
6300 LOCATE 22,25: PRINT "
"
6310 LOCATE 21,18: PRINT "ENTER 1 TO REPEAT THIS LAB WITH
THE NEXT AOA"
6320 LOCATE 22,20: INPUT "ENTER 2 TO EXIT BASICA AND RETURN
TO DOS";AN20
6330 IF AN20 <> 2 GOTO 1430 ELSE 5180
6340 STOP
6350 '
6360 '
6370 ' SUBROUTINE FOR READING AOA
6380 '
6390 '
6400 INPUT.CH = 5
6410 CALL OUTPUT(RELAY.MUX.01,INPUT.CH)
6420 CALL ENABLE.OUTPUT(RELAY.MUX.01)
6430 TIME = .5
6440 CALL DELAY(TIME)
6450 CALL MEASURE(DMM.01,AOA)
6460 IF AOA < 0 GOTO 6470 ELSE 6490
6470 HAVEAOA = (AOA-.003)*1000/10
6480 GOTO 6500
6490 HAVEAOA = (AOA-.003)*1000/10
6500 RETURN
6510 '
6520 '
6530 ' SUBROUTINE FOR AOA SELECTION
6540 '
6550 '
6560 WHILE (CINT(ABS(WANTAOA - HAVEAOA) * 10) / 10) >= .25
6570   ABSDIFF = ABS(WANTAOA - HAVEAOA)
6580   IF ABSDIFF > 36 THEN ABSDIFF = 36
6590   SWITCH.1 = 1
6600   SWITCH.2 = 2
6610   CALL OPEN.CHANNEL(RELAY.ACT.01,SWITCH.1)
6620   CALL OPEN.CHANNEL(RELAY.ACT.01,SWITCH.2)
6630   IF WANTAOA > HAVEAOA GOTO 6670 ELSE 6800
6640   '
6650   ' THIS PART OF THE SUBROUTINE MOVES THE TURNTABLE
FORWARD
6660   '
6670   IF (CINT(ABSDIFF * 10) / 10) >= .4 THEN TIMING = 100
ELSE TIMING = 1
```

```
6680   COUNT = ((ABSDIFF * TIMING) + 190) / 1000
6690   IF COUNT > 3 THEN COUNT = 3
6700   CALL CLOSE.CHANNEL(RELAY.ACT.01,SWITCH.1)
6710   CALL DELAY(COUNT)
6720   CALL OPEN.CHANNEL(RELAY.ACT.01,SWITCH.1)
6730   TIME = .5
6740   CALL DELAY(TIME)
6750   GOSUB 6400
6760   GOTO 6900
6770   '
6780   ' THIS PART OF THE SUBROUTINE MOVES THE TURNTABLE IN
REVERSE
6790   '
6800   IF (CINT(ABSDIFF * 10) / 10) >= .4 THEN TIMING = 100
ELSE TIMING = 1
6810   COUNT = ((ABSDIFF * TIMING) + 190) / 1000
6820   IF COUNT > 3 THEN COUNT = 3
6830   CALL CLOSE.CHANNEL(RELAY.ACT.01,SWITCH.2)
6840   CALL DELAY(COUNT)
6850   CALL OPEN.CHANNEL(RELAY.ACT.01,SWITCH.2)
6860   TIME = .5
6870   CALL DELAY(TIME)
6880   GOSUB 6400
6890   GOTO 6900
6900 WEND
6910 RETURN
6920 '
6930 '
6940 ' SUBROUTINE FOR MEASURING WIND TUNNEL TEMPERATURE
6950 '
6960 '
6970 COLOR 1
6980 A1 = 38.709457#
6990 A2 = .037085566#
7000 A3 = 5.649552E-05
7010 B0 = .10086091#
7020 B1 = 25727.94369#
7030 B2 = -767345.8295#
7040 B3 = 78025595.81#
7050 B4 = -9247486589#
7060 B5 = 6.97688E+11
7070 B6 = -2.66192E+13
7080 B7 = 3.94078E+14
7090 B8 = 0
7100 B9 = 0
7110 REF.OUT = 8
7120 CALL OUTPUT(RELAY.MUX.01,REF.OUT)
7130 TIME = .5
7140 CALL DELAY(TIME)
7150 CALL MEASURE(DMM.01,V)
7160 TR = 100 * V
```

```
7170 ER = (TR*(A1+TR*(A2+TR*A3)))*10^-6
7180 CALL SET.RANGE(DMM.01,R200MILLI)
7190 TC = 7
7200 CALL OUTPUT(RELAY.MUX.01,TC)
7210 TIME = .5
7220 CALL DELAY(TIME)
7230 CALL MEASURE(DMM.01,ET)
7240 E = ER + ET
7250 Z = B5+E*(B6+E*(B7+E*(B8+E*B9)))
7260 T = B0+E*(B1+E*(B2+E*(B3+E*(B4+E*Z))))
7270 TF = (1.8 * T) + 32
7280 RETURN
7290 '
7300 '
7310 ' SUBROUTINE FOR WIND TUNNEL DATA:
7320 '              STATIC PRESS IN TEST SECTION
7330 '              AIR DENSITY IN TEST SECTION
7340 '              DYNAMIC PRESSURE IN TEST SECTION
7350 '              TEST SECTION VELOCITY, FT/SEC AND KTS
7360 '
7370 '
7380 CLS: COLOR 1: LOCATE 10,15: PRINT "DEPRESS BUTTON    ON
THE PARASCIENTIFIC TRANSDUCER"
7390 COLOR 4: LOCATE 10,30: PRINT "P2": COLOR 1
7400 LOCATE 12,15: PRINT "MAKE SURE NO OTHER FUNCTION
BUTTONS ARE DEPRESSED"
7410 COLOR 4: LOCATE 21,28: PRINT "PRESS ENTER TO CONTINUE"
7420 A$ = INKEY$: IF A$ = "" THEN 7420
7430 I = 0: SUM3 = 0
7440 FOR I = 1 TO 10
7450   CALL MEASURE(DIG.IN.01,J(I))
7460   SUM3 = SUM3 + J(I)
7470 NEXT I
7480 DN = SUM3/10
7490 GOSUB 7900
7500 PSTAT = PRESS * 144
7510 IF PSTAT < 1000 THEN RETURN
7520 TTEMP = TF + 459.67
7530 GASCONST = 53.3
7540 DENSITY = PSTAT/(GASCONST * TTEMP)
7550 CLS: COLOR 1: LOCATE 10,15: PRINT "DEPRESS BUTTON    ON
THE PARASCIENTIFIC TRANSDUCER"
7560 COLOR 4: LOCATE 10,30: PRINT "P1": COLOR 1
7570 LOCATE 12,15: PRINT "MAKE SURE NO OTHER FUNCTION
BUTTONS ARE DEPRESSED"
7580 COLOR 4: LOCATE 21,28: PRINT "PRESS ENTER TO CONTINUE"
7590 A$ = INKEY$: IF A$ = "" THEN 7590
7600 I = 0: SUM1 = 0
7610 FOR I = 1 TO 70
7620   CALL MEASURE(DIG.IN.01,F(I))
7630   F(I) = F(I) + .007
```

```
7640   SUM1 = SUM1 + F(I)
7650 NEXT I
7660 DN = SUM1/70
7670 GOSUB 7900
7680 VELOC = PRESS
7690 IF VELOC > 10 THEN VELOC = 0
7700 DELTAP = VELOC*144
7710 Q = DELTAP/.93
7720 IF Q < 1 THEN Q = 0
7730 GC = 32.174
7740 VEL = SQR(ABS((2*Q*GC)/DENSITY))
7750 KTS = VEL * .5924
7760 FILE$ = "HOTWIRE.HPC"
7770 CALL INITIALIZE.SYSTEM(FILE$)
7780 CALL ENABLE.SYSTEM
7790 CALL SET.FUNCTION(DMM.01,DCVOLTS)
7800 CALL SET.RANGE(DMM.01,AUTOM)
7810 CALL SET.SPEED(DMM.01,R2.5)
7820 CALL DISABLE.INT.TRIGGER(DMM.01)
7830 CALL ENABLE.OUTPUT(RELAY.MUX.01)
7840 CALL SET.NUM.BITS(DIG.IN.01,R16)
7850 LEVEL = 5
7860 CALL SET.THRESHOLD(DIG.IN.01,LEVEL)
7870 RETURN
7880 '
7890 '
7900 ' SUBROUTINE FOR CONVERTING DECIMAL PRESSURE VALUES TO
NUMERIC VALUES
7910 '
7920 '
7930 COUNT = 0: PRESS = 0
7940 '
7950 ' DN IS THE MEASURED DECIMAL NUMBER
7960 ' DEC IS THE DECIMAL TRANSFORM MATRIX
7970 ' PRESS IS THE CONVERTED NUMERIC PRESSURE VALUE
7980 '
7990 DEC(16) = 10: DEC(15) = 8: DEC(14) = 4: DEC(13) = 2:
DEC(12) = 1
8000 DEC(11) = .8: DEC(10) = .4: DEC(9) = .2: DEC(8) = .1:
DEC(7) = .08
8010 DEC(6) = .04: DEC(5) = .02: DEC(4) = .01: DEC(3) =
8.000001E-03
8020 DEC(2) = .004: DEC(1) = .002
8030 '
8040 IF DN < 0 THEN PRESS = DEC(16) ELSE GOTO 8060
8050 COUNT = -32768!
8060 IF (DN-COUNT)/16384 >= 1 THEN PRESS = PRESS + DEC(15)
ELSE GOTO 8080
8070 COUNT = COUNT + 16384
8080 IF (DN-COUNT)/8192 >= 1 THEN PRESS = PRESS + DEC(14)
ELSE GOTO 8100
```

```
8090 COUNT = COUNT + 8192
8100 IF (DN-COUNT)/4096 >= 1 THEN PRESS = PRESS + DEC(13)
ELSE GOTO 8120
8110 COUNT = COUNT + 4096
8120 IF (DN-COUNT)/2048 >= 1 THEN PRESS = PRESS + DEC(12)
ELSE GOTO 8140
8130 COUNT = COUNT + 2048
8140 IF (DN-COUNT)/1024 >= 1 THEN PRESS = PRESS + DEC(11)
ELSE GOTO 8160
8150 COUNT = COUNT + 1024
8160 IF (DN-COUNT)/512 >= 1 THEN PRESS = PRESS + DEC(10)
ELSE GOTO 8180
8170 COUNT = COUNT + 512
8180 IF (DN-COUNT)/256 >= 1 THEN PRESS = PRESS + DEC(9) ELSE
GOTO 8200
8190 COUNT = COUNT + 256
8200 IF (DN-COUNT)/128 >= 1 THEN PRESS = PRESS + DEC(8) ELSE
GOTO 8220
8210 COUNT = COUNT + 128
8220 IF (DN-COUNT)/64 >= 1 THEN PRESS = PRESS + DEC(7) ELSE
GOTO 8240
8230 COUNT = COUNT + 64
8240 IF (DN-COUNT)/32 >= 1 THEN PRESS = PRESS + DEC(6) ELSE
GOTO 8260
8250 COUNT = COUNT + 32
8260 IF (DN-COUNT)/16 >= 1 THEN PRESS = PRESS + DEC(5) ELSE
GOTO 8280
8270 COUNT = COUNT + 16
8280 IF (DN-COUNT)/8 >= 1 THEN PRESS = PRESS + DEC(4) ELSE
GOTO 8300
8290 COUNT = COUNT + 8
8300 IF (DN-COUNT)/4 >= 1 THEN PRESS = PRESS + DEC(3) ELSE
GOTO 8320
8310 COUNT = COUNT + 4
8320 IF (DN-COUNT)/2 >= 1 THEN PRESS = PRESS + DEC(2) ELSE
GOTO 8340
8330 COUNT = COUNT + 2
8340 IF (DN-COUNT) = 1 THEN PRESS = PRESS + DEC(1)
8350 RETURN
8360 '
8370 '
8380 ' SUBROUTINE FOR INTEGRATION
8390 '            INPUTS:   X(I), Y(I)  TABLE VALUES IN
"PART2.DAT"
8400 '                      V = NUMBER OF TABLE VALUES
8410 '                      X1, X2  INTEGRATION RANGE
8420 '                      Z = CALCULATED INTEGRAL
8430 '
8440 '
8450 V = NUMBER
8460 GOSUB 9060
```

```
8490 RETURN
8500
'************************************************************
*****
8510 '              AKIMA SPLINE FITTING SUBROUTINE
8520 '   THE INPUT TABLE IS (X(I), Y(I)), WHERE Y(I) IS THE
DEPENDENT VARIABLE
8530 '   THE INTERPOLATION POINT IS X, WHICH IS ASSUMED TO
BE IN THE RANGE OF
8540 '   THE TABLE WITH AT LEAST ONE TABLE POINT TO THE
LEFT, AND THREE TO THE
8550 '   RIGHT. Y IS RETURNED AS THE INTERPOLATED VALUE. N
IS RETURNED AS AN
8560 '   ERROR CHECK (N = 0 IMPLIES AN ERROR). DIMENSION
M,X,Y, AND Z IN THE
8570 '   CALLING PROGRAM.
8580 N = 1
8590 ' CHECK TO SEE IF X IS IN THE TABLE RANGE
8600 IF X > = X(1) GOTO 8630
8610 N = 0
8620 RETURN
8630 IF X <= X(V-3) GOTO 8660
8640 N = 0
8650 RETURN
8660 X(0) = 2*X(1)-X(2)
8670 ' CALCULATE AKIMA COEFFICIENTS
8680 I = 0
8690 FOR I = 1 TO V-1
8700  ' SHIFT I TO I+2
8710   M(I+2) = (Y(I+1)-Y(I))/(X(I+1)-X(I))
8720 NEXT I
8730 M(V+2) = 2*M(V+1) - M(V)
8740 M(V+3) = 2*M(V+2) - M(V+1)
8750 M(2) = 2*M(3) - M(4)
8760 M(1) = 2*M(2) - M(3)
8770 I = 0
8780 FOR I = 1 TO V
8790   A = ABS(M(I+3) - M(I+2))
8800   B = ABS(M(I+1) - M(I))
8810   IF A+B <> 0 GOTO 8840
8820   Z(I) = (M(I+2) + M(I+1))/2
8830   GOTO 8850
8840   Z(I) = (A*M(I+1) + B*M(I+2))/(A+B)
8850 NEXT I
8860 ' FIND RELEVANT TABLE INTERVAL
8870 I = 0
8880 I = I + 1
8890 IF X >= X(I) GOTO 8880
8900 I = I - 1
8910 ' BEGIN INTERPOLATION
8920 B = X(I+1) - X(I)
```

```
8930 A = X - X(I)
8940 Y = Y(I)+Z(I)*A+(3*M(I+2)-2*Z(I)-Z(I+1))*A*A/B
8950 Y = Y+(Z(I)+Z(I+1)-2*M(I+2))*A*A*A/(B*B)
8960 RETURN
8970 '
****************************************************************
****
8980 ' GENERAL INTEGRATION SUBROUTINE (ITEG)
8990 '    INTERPOLATION BY AKIMA
9000 '    INTEGRATION BY ENHANCED TRAPEZOIDAL RULE
9010 '    THE INTEGRATION RANGE IS (X1, X2).
9020 '    IT IS ASSUMED THAT X1 < X2, AND THAT THERE IS AT
LEAST ONE TABLE
9030 '    VALUE TO THE LEFT OF X1, AND THREE TO THE RIGHT OF
X2.
9040 '    THE RESULT IS RETURNED IN Z.
9050 '    AN ERROR CHECK IS RETURNED IN Z1. Z1 = 0 IMPLIES
ERROR.
9060 Z = 0
9070 Z1 = 0
9080 ' CHECK TO SEE IF END POINTS ARE IN ALLOWABLE RANGE
9090 IF X1 < X(1) THEN RETURN
9100 IF X2 > X(V-3) THEN RETURN
9110 ' IF X1 > X2 THEN SWITCH AND SET FLAG
9120 IF X1 < X2 GOTO 9170
9130 X3 = X1
9140 X1 = X2
9150 X2 = X3
9160 Z1 = 1
9170 IF X2 = X1 THEN RETURN
9180 ' START TRAPEZOIDAL INTEGRATIONS
9190 ' FIRST INTEGRATION TO GET I1
9200 GOSUB 9380
9210 ' SECOND ROUND TO GET I2
9220 GOSUB 9680
9230 ' RICHARDSON EXTRAPOLATION
9240 Z = 4*I2/3 - I1/3
9250 ' CHECK TO SEE IF THE END POINTS HAVE BEEN REVERSED
9260 IF Z1 = 0 GOTO 9310
9270 Z = -Z
9280 X2 = X1
9290 X1 = X3
9300 ' RESET ERROR FLAG
9310 Z1 = 1
9320 RETURN
9330 ' Z IS THE INTEGRAL DESIRED
9340 RETURN
9350 '
****************************************************************
*
9360 ' ROUTINE FOR THE FIRST TRAPEZOIDAL INTEGRATION, I1
```

```
9370 '  N1 KEEPS TRACK OF THE NUMBER OF INTERVALS
9380 I1 = 0
9390 N1 = 0
9400 X = X1
9410 ' FIND THE BEGINING OF THE INTERVAL
9420 ' GO TO BRANCH WHICH CALLS THE INTERPOLATION ROUTINE
9430 ' FIND THE INTERVAL, I, AND THE LEFT END POINT, Y.
9440 GOSUB 9990
9450 IF X2 > X(I+1) GOTO 9540
9460 N1 = N1 + 1
9470 D = Y
9480 X = X2
9490 ' FIND END POINT Y VALUE
9500 GOSUB 9990
9510 I1 = (Y+D)*(X2-X1)/2
9520 RETURN
9530 ' AT LEAST ONE TABLE INTERVAL MUST BE SUMMED OVER
9540 J1 = I
9550 I1 = I1 + (Y+Y(I+1))*(X(I+1)-X)/2
9560 IF X2 < X(J1+3) GOTO 9610
9570 N1 = N1 + 1
9580 I1 = I1+(Y(J1+1)+Y(J1+3))*(X(J1+3)-X(J1+1))/2
9590 J1 = J1 + 2
9600 GOTO 9560
9610 X = X2
9620 GOSUB 9990
9630 I1 = I1+(Y+Y(J1+1))*(X2-X(J1+1))/2
9640 N1 = N1 + 1
9650 RETURN
9660 '
************************************************************
***
9670 ' INTEGRATION FOR I2
9680 I2 = 0
9690 X = X1
9700 GOSUB 9990
9710 D = Y
9720 IF X2 > X(I+1) GOTO 9810
9730 X = X1+(X2-X1)/2
9740 GOSUB 9990
9750 I2 = I2+(D+Y)*(X2-X1)/4
9760 D = Y
9770 X = X2
9780 GOSUB 9990
9790 I2 = I2+(D+Y)*(X2-X1)/4
9800 RETURN
9810 X = X1+(X(I+1)-X1)/2
9820 J1 = I
9830 GOSUB 9990
9840 I2 = I2+(Y+D)*(X-X1)/2
9850 I2 = I2+(Y+Y(J1+1))*(X(J1+1)-X)/2
```

```
9860 IF X2 < X(J1+2) GOTO 9900
9870 I2 = I2+(Y(J1+1)+Y(J1+2))*(X(J1+2)-X(J1+1))/2
9880 J1 = J1 + 1
9890 GOTO 9860
9900 X = X2-(X2-X(J1+1))/2
9910 GOSUB 9990
9920 D = Y
9930 I2 = I2+(Y(J1+1)+D)*(X2-X)/2
9940 X = X2
9950 GOSUB 9990
9960 I2 = I2+(D+Y)*(X2-X(J1+1))/4
9970 RETURN
9980 ' GO TO AKIMA SPLINE INTERPOLATION SUBROUTINE
9990 GOSUB 8580
10000 ' JUST RETURNED FROM INTERPOLATION SUBROUTINE
10010 ' RETURN TO PROGRAM
10020 RETURN
10030 END
```

LIST OF REFERENCES

1. Ruckdeschel, F.R.,
Basic Scientific Subroutines Vol. II, pp. 316-325,
McGraw-Hill Book Co., 1981.

2. Laboratory Manual for Low Speed Wind Tunnel Testing,
Department of Aeronautics, Naval Postgraduate School,
Monterey, California, October 1983

BIBLIOGRAPHY

Abbott,I.H. and Von Doenhoff,A.E.,
Theory of Wing Sections,
McGraw-Hill Book Co., 1949

Goldstein, R.J.,
Fluid Mechanics Measurements,
Hemisphere Publishing Corporation, 1983

Pope, A. and Harper, J.J.,
Low Speed Wind Tunnel Testing,
John Wiley & Sons, Inc., 1966

TSI Technical Bulletin 18,
Procedure for Temperature Correction on Velocity Measurements

INITIAL DISTRIBUTION LIST

		NO. COPIES
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5000	2
3.	Satya Bodapati, Code 67 Bu Department of Aeronautics Naval Postgraduate School Monterey, California 93943-5000	10
4.	M. F. Platzer, Chairman Department of Aeronautics Naval Postgraduate School Monterey, California 93943-5000	1
5.	Gregory J. Doremus 3 Birchbrook Drive Smithtown, New York 11787	5
6.	Jack King Department of Aeronautics, Code 67Jk Naval Postgraduate School Monterey, California 93943-5000	1
7.	Alan McGuire Department of Aeronautics, Code 67Am Naval Postgraduate School Monterey, California 94343-5000	1
8.	D. P. Bencze Chief, Applied Aerodynamics Branch NASA Ames Research Center (M. S. 227-6) Moffett Field, California 94035	1
9.	Chief, Fluid Mechanics Laboratory NASA Ames Research Center (M.S. 260-1) Moffett Field, California 94035	1
10.	Engineering Sciences Divison Attn: T. L. Doligalski, Chief, Fluid Dynamics U.S. Army Research Office P. O. Box 12211 Research Triange Park North Carolina 27709-2211	1

11. Directorate of Aerospace Sciences 1
Attn: Hank E. Helin, Program Manager
AFOSR/NA
Bolling Air Force Base
Washington, DC 20332-6448

12. Naval Air Systems Command 1
Attn: Harry Berman, Program Manager
AIR-931K
Washington, D.C. 20361-9320

13. Office of Naval Research 1
Attn: Spiro Lekoudis, Program Manager
Code 1132F
800 North Quincy Street
Arlington, Virginia 22217-5000

14. David Taylor, NSRDC, Code 8 1
Attn: Jim Nichols
Aviation and Surface Effects Department
Bethesda, Maryland 20084

Thesis
D6535 Doremus
c.1 Flow field measurements
using hotwire anemometry.